INTEGRATING VISUAL AND SEMANTIC DESCRIPTIONS FOR EFFECTIVE, FLEXIBLE AND USER-FRIENDLY IMAGE RETRIEVAL

THÈSE N° 2679 (2002)

PRÉSENTÉE À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS

SECTION DES SYSTÈMES DE COMMUNICATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Zoran PECENOVIC

ingénieur en informatique diplômé EPF
et de nationalité croate

acceptée sur proposition du jury:

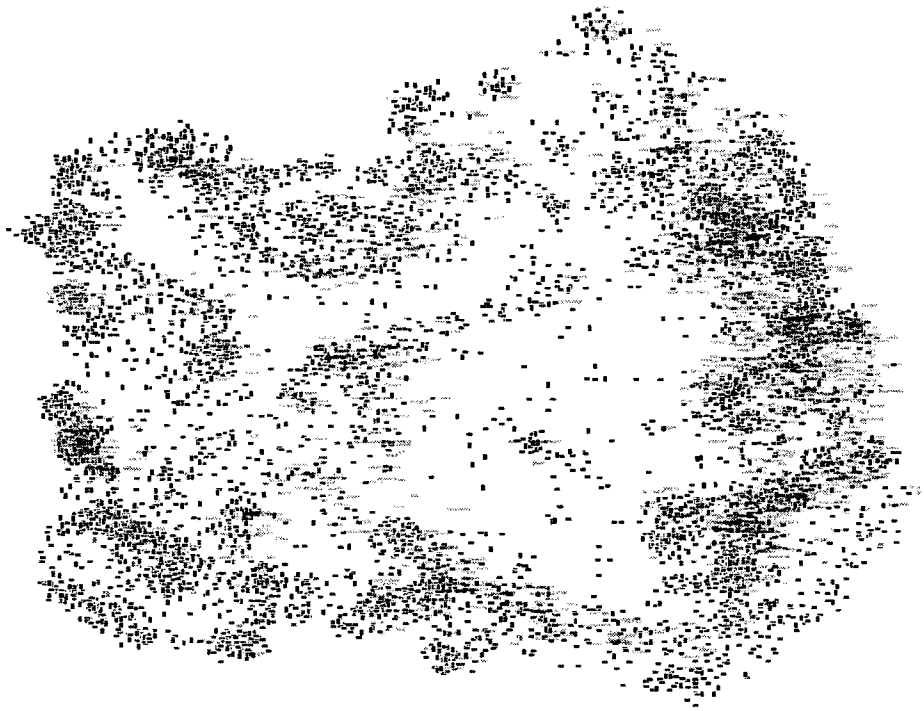
Prof. M. Vetterli, directeur de thèse
Dr S. Ayer, rapporteur
Prof. R. Leonardi, rapporteur
Dr P. Pu. Faltings, rapporteur
Prof. S. Süsstrunk, rapporteur

Lausanne, EPFL
2003

Integrating visual and semantic descriptions for effective, flexible and user-friendly image retrieval

Zoran Pečenović

September, 2002



Abstract

Effectively managing a large collection of multimedia documents is a challenge, addressed by many disciplines from signal processing through database systems to artificial intelligence and interaction design. The problems to be solved have rarely been considered together. We propose a series of novel solutions for: the system architecture, the document content characterization, the retrieval methodology, and the user interaction schemes.

We propose and describe a communicating components architecture using a new open and flexible protocol for messaging. Its foundation, the multimedia retrieval markup language (`mrml`) is specified with examples of the benefits of this approach.

The general multimedia application domain is restricted mainly to image documents that carry semantic annotation like captions or meta-data. The integration of perceptual and semantic content descriptions into a single retrieval structure is the principal contribution of our work. The proposed method allows for improved effectiveness, augmented functionality and high flexibility for the extension to other media types like audio or video. The relationships among the two different representational characteristics can be exploited for more effective retrieval and other tasks like semi-automatic image annotation or illustration of semantic concepts.

The final contribution is the study and implementation of user interaction metaphors that are intuitive and attempt to further bridge the gap between user's and system's notion of relevance. A rich set of information need formulation tools is proposed to users of varying skill and of varying requirements. Merging query specification with result visualization and browsing, using interactively explorable search spaces, offers a single access point to most retrieval tasks.

Résumé

La gestion satisfaisante d'une vaste collection de documents multimédia est toujours un défi. Plusieurs domaines de recherche y ont contribué, le traitement numérique des signaux, les bases de données, l'intelligence artificielle ou encore l'ergonomie et l'interaction homme-machine. Les problèmes encore ouverts n'ont été que rarement considérés en leur intégrité. Nous proposons une étude qui fournit une série de solutions nouvelles pour : l'architecture du système, pour la caractérisation du contenu des documents, pour la méthode de recherche et pour les modalités d'interaction entre utilisateur et système.

Nous présentons une architecture en composants communicants qui utilisent un protocole nouveau, flexible et ouvert. Le fondement de ce protocole, le `mrml` (multimedia retrieval markup language) est spécifié et accompagné d'exemples qui témoignent des bénéfices encourus en adoptant cette approche.

Le domaine général des applications aux documents multimédia a été restreint aux documents images associés avec des descriptions sémantiques telles que légendes, annotations ou meta-donnés. La contribution principale de notre travail est l'intégration de descriptions du contenu perceptuelles avec la sémantique dans une seule structure dédiée à l'interrogation. La méthode proposée, par rapport aux méthodes courantes, offre des performances accrues, des fonctionnalités supplémentaires et une très grande flexibilité notamment pour l'extension aux documents de divers types de médias. Les relations entre ces deux types de représentations du contenu peuvent être identifiées et exploitées pour une interrogation plus précise, ainsi que pour des tâches nouvelles comme l'annotation semi-automatique d'images ou l'illustration de concepts sémantiques.

La contribution finale de nos investigations est l'étude et l'implémentation de métaphores d'interaction intuitives qui tentent de joindre et harmoniser la représentation de pertinence de l'utilisateur avec celle du système. Nous proposons un vaste ensemble d'outils pour la formulation des besoins en informations, adaptés aux aptitudes et aux exigences des différents utilisateurs. La fusion de la spécification des requêtes avec la visualisation des résultats, en utilisant des espaces de recherche interactivement explorables, offre un point unique d'accès à la plupart des opérations de recherche.

Acknowledgments

I would like to thank:

My supervisor, Martin Vetterli, for the continuous and precious advice, the patience with my "multi-resolution" editing, the motivation and the support.

My co-supervisor, Pearl Pu, for her counsel and enriching ideas, for her understanding, encouragement and support.

Laurent Balmelli for his shrewd comments and careful reading of the drafts.

My colleagues, Minh, Pier, and office mates George, Abdel and Guillermo for the nice times and for the stimulating discussions.

Wolfgang Müller, Thierry Pun, and the rest of the Geneva team for the discussions and collaboration.

My friends, for their support, care and friendship, and especially Antonio, Laurent, Matthias and Sandro (in alphabetical order).

My family for their love, attention and support.

And finally, Cristina, for much of the above and all the rest.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 What is multimedia retrieval?

The technological, cultural and economic changes that came about with the advent of the information era, are rich in potential benefits, but still present a number of challenges. The information is abundant, it is in many cases accessible and available, but its sheer volume and diversity make it difficult to pinpoint the actual useful material. Knowledge, stored in books or transmitted by oral tradition, has long ago outgrown the human synthetic capacity. Paper-based catalogs, indexes, and digests can still be useful but with today's digital technology we are getting closer to building an ideal information "assistant".

The central ability of such an assistant (called system later on) is multimedia retrieval. In simple terms, a user explains to the system what the needed information is and the assistant's role is to find it, structure it and present it to the user. The results of such an interaction consist of a set of digital documents that contain the desired information. The medium that would convey this information could be textual, still image, video, audio or a free mixture thereof. This polymorphic nature requires flexible need formulation and effective information-storage, -access and -transmission strategies.

These individual sub-problems have already been studied. The domain of multimedia retrieval is at a meeting point of different research areas: databases, signal processing, communications, artificial intelligence, human-computer interaction and many more (Figure 1.1).

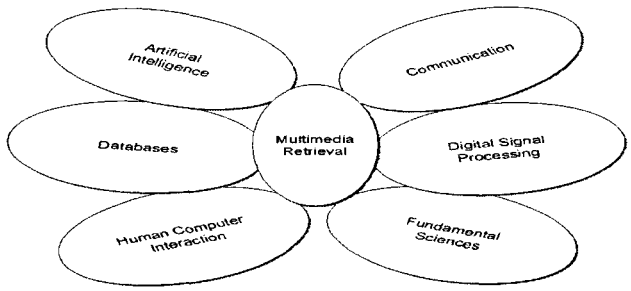


FIGURE 1.1: Multimedia retrieval: a meeting point

Since we are facing a new discipline, this distribution of efforts is natural, enlightening and brings alternative views to common problems. However, as Meghini *et al.* (2001) point out: "on the long run, this very richness may ultimately result in a fragmentation of efforts that may slow down progress." For this reason, and since we are convinced that a complete framework for multimedia retrieval is necessary, we propound an integrated investigation of the essential sub-problems. Figure 1.2 illustrates the basic components of a complete multimedia retrieval system. Each one could present sufficient challenges and many open issues, but our goal is to provide an all-encompassing investigation to these problems.

Even though the more generic principles discussed in this thesis apply to information contained in documents that intermix text, image, video and sound, we concentrate and develop primarily the image+text document case. In this kind of document, the conspicuous part is an image, the text being a descriptive caption. The converse situation of a significant body of text illustrated by images can also be treated with minimal modifications. Video and audio data, entailing drastically different user

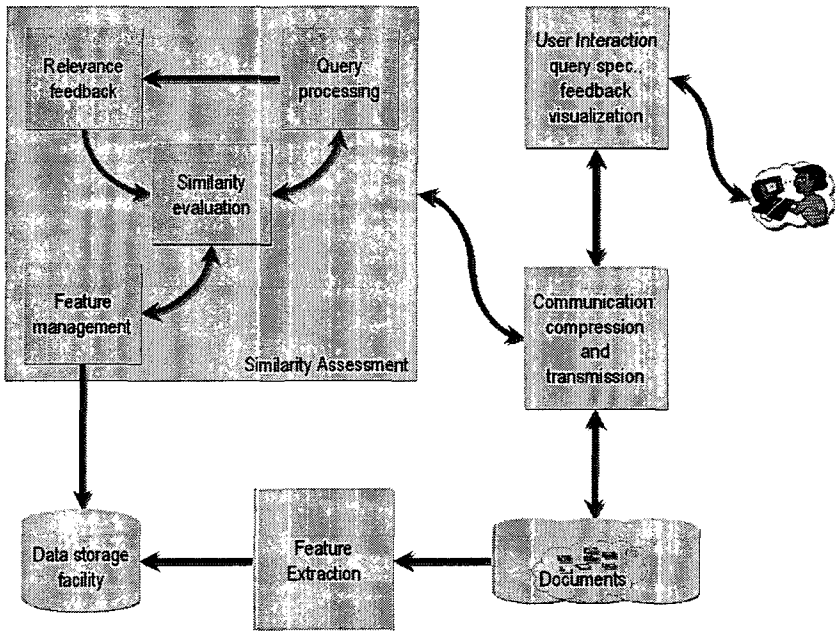


FIGURE 1.2: Essential components of a multimedia retrieval system.

objectives and approaches, are not treated in any detail in this work, although the aural characteristics of music or speech, and the intrinsic time-varying characteristics of video can be accommodated with the addition of the appropriate feature extraction and management components.

We have however, concentrated our investigation on three aspects that span somewhat the various components. The major contributions of our work can be summarized by the following points:

- Seamless and transparent integration of textual and visual cues for annotated image retrieval.
- Implementation of intuitive, ergonomic, interactive and efficient methods for many retrieval tasks.
- Development of a framework and communication protocol for complex multimedia-retrieval systems.

1.2 Is a word better than a thousand images?

A large part of this work is dedicated to the unification of visual and semantic cues for image+text document retrieval. The cliché:

A picture is worth a thousand words,

has turned out to be quite difficult to prove in multimedia retrieval. The efforts devoted to the refinement of retrieval based on visual similarity by example — ranging from the initial works by Hirata and Kato (1992) or Faloutsos *et al.* (1994) to the more recent developments surveyed in (Smeulders *et al.*, 2000) — have often produced only barely convincing results for the layman. Converse approaches, that tackle content-based retrieval by semantic annotation, have also been deemed insufficient. While the former suffer from as yet unsolved computational, representational and analytical problems, the latter do so mainly due to poor or inadequate semantic and content descriptive data associated to the image documents.

The obvious, yet rarely addressed, solution is the amalgamation of the two approaches (Figure 1.3). The benefits of integrating both aspects are essentially:

Increase of effectiveness Relationships between visual and semantic characteristics of a document can be exploited to retrieve non annotated or not visually similar yet relevant material. Automatic or assisted semantic annotation of the data can be provided and results can be associated with explanatory cues for their relevance.

Increase of functionality Previously studied means of formulating information needs can be complemented with novel combined textual and visual queries and powerful result visualizations. The nature of the constructs used for retrieval can be kept hidden from the user of the system, without compromising effectiveness or user satisfaction.

FIGURE 1.3: The associations between visual and semantic content.

1.3 A case for intelligent interaction models

The manner in which the user interacts with the multimedia retrieval system should be closely linked to the type of media that is to be retrieved. The natural specification of the information need and the subsequent examination of the returned results should be tailored not only to the application domain and user specifics, but also to the information that is being communicated. In our restricted application to image+text documents we share the view of Shneiderman (1983) proposing direct manipulation, i.e., providing maximum control to the user. However, this should be done only when explicitly requested, maintaining simple interaction paths for users not familiar with information retrieval systems.

The basic steps of the user's task in information retrieval consist of i) formulating the information need, ii) instructing the system to return relevant information, and iii) examining the results. Eventually, these steps can repeat with modifications to the need formulation until satisfactory results are retrieved. We complement this basic interaction model with other more immersive approaches where the three steps are not so clearly separated. The query formulation and result visualization can be performed simultaneously, in the same visual environment. Similarly the user can be guided to comprehend the system's viewpoint by showing him, along with the relevant material, an understandable representation of the reasons why the system thought the results were relevant.

Figure 1.4 shows the display of an interactive session where the user navigates through a representation of the document collection, where the material relevant to the current query is visually highlighted.

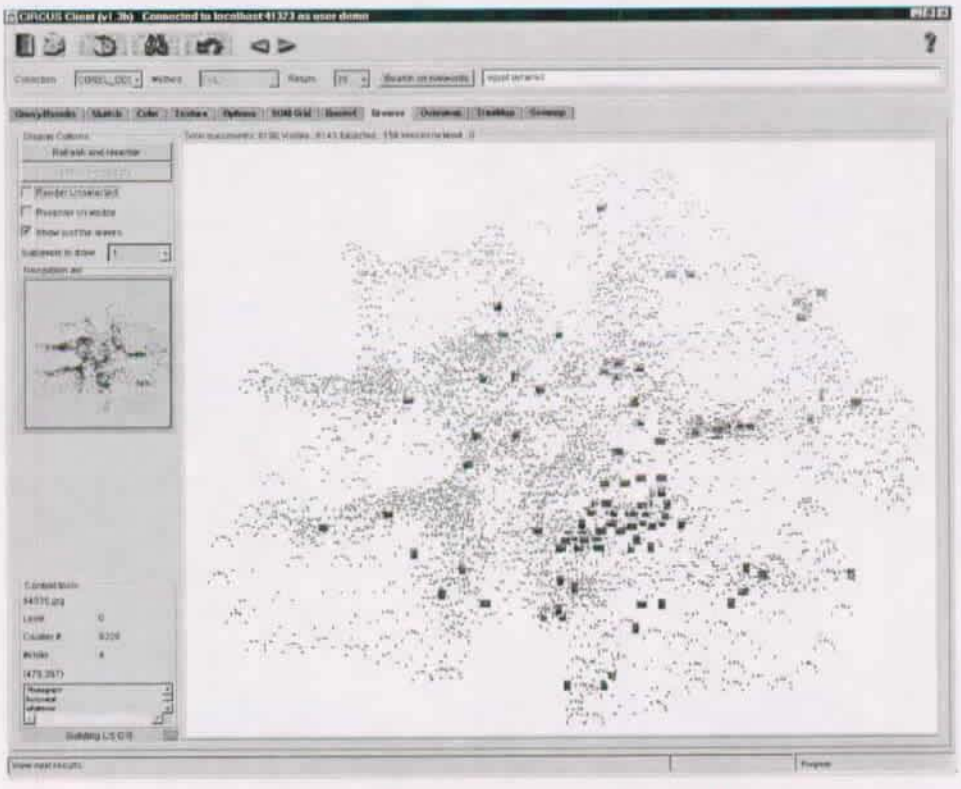


FIGURE 1.4: The immersive interaction model.

Alternatively, the user can compare the effects of using different methods or criteria for retrieval. This trade-off space allows the user to select the more appropriate method for future queries, or queries

in a different context, and, coupled with the previous immersive interaction model, guides her/him to more effective information need formulations (Figure 1.5).

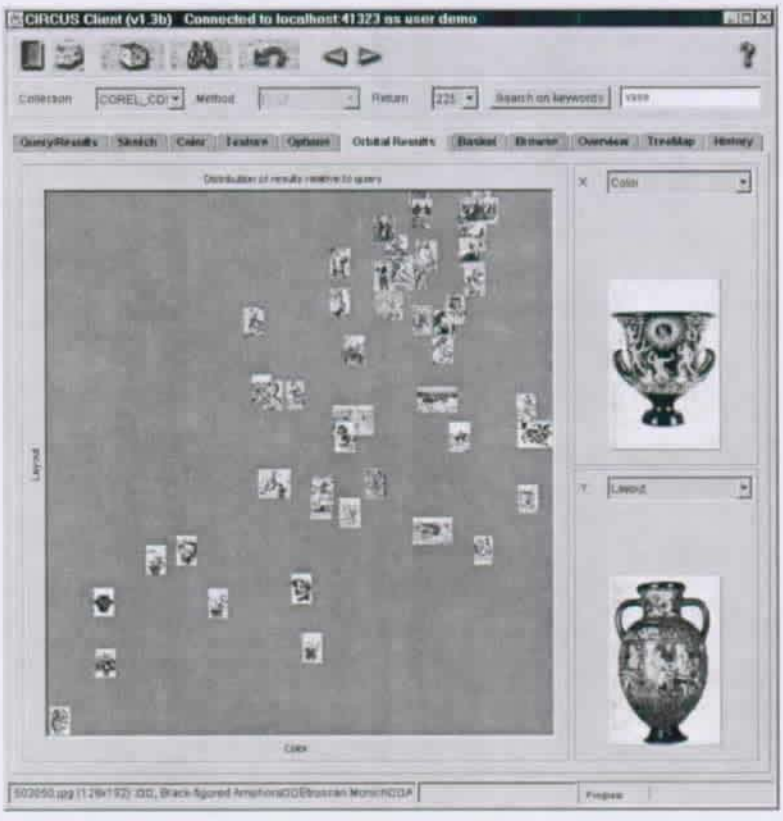


FIGURE 1.5: Trade-off space interaction model.

1.4 Managing system complexity

The entities presented in Figure 1.2 are more or less complex data processing units that can produce equivalent results in many different ways. The coordination and communication between these software components (the arrows on the schema) should be managed with a general mechanism that permits substitution of equivalent units, without changing the configuration of the rest of the system. The meaning of the processed data and the driving parameters should be made comprehensible for both the human system manager or user and the software components. These concerns lead us to the formulation of a protocol for component communication, with the following primordial characteristics: flexible, extensible, effective and self-explanatory.

The distributed nature of today's large, heterogeneous knowledge- and data-bases, requires a protocol that can span local and wide area networks, offering remote users all the retrieval engine's functionality. The number of information sources containing the desired documents is likely large, and the user should be able to communicate with all of them through a single, simple access point. A situation that illustrates these scenarios is depicted on Figure 1.6.

These internal communication issues of the multimedia retrieval problem are also treated in some detail in this thesis. The problem of data compression and transmission, especially of image, audio or video content, is only briefly reviewed, and is not considered as a central issue in this work.

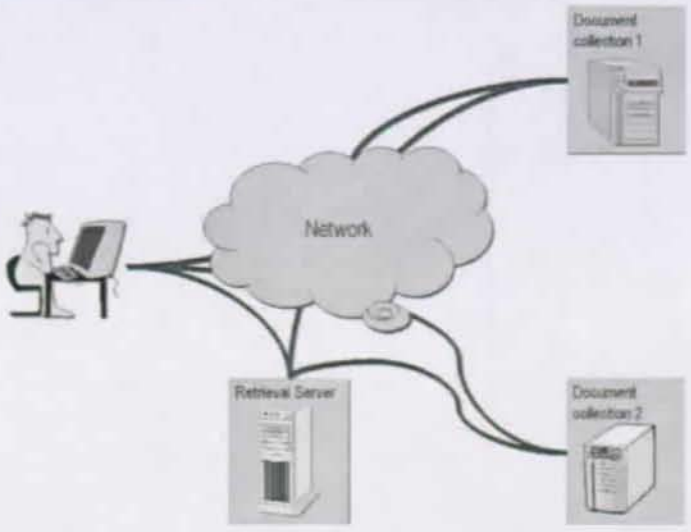


FIGURE 1.6: Distributed setup for information systems.

1.5 Road map

The structure of the dissertation is tailored to this systemic view of the problem, where each chapter presents one of the major sub-problems. The exposition order is the following:

Chapter 2 details the architecture of the proposed framework for multimedia retrieval, it surveys the previous approaches and presents a new view of the system as inter-connected processing components.

Next, Chapter 3 presents the communication protocol and associated message formatting language that permits the inter-operability of the other components. This open standard is a further contribution, resulting from collaboration with the Vision and Multimedia Lab of the University of Geneva (Marchand-Maillet, 1997—). It also presents some results obtained using this protocol with different image retrieval systems.

Chapter 4 is dedicated to the basic issues of characterizing image visual content. We present several contributions to the *feature extraction* problem, but our goal was not to create new, advanced models of image description, but rather to build a sturdy basis for a more involved, and we believe, more interesting problem: the merging of visual and semantic content.

Chapter 5 is an intermezzo, presenting a retrieval method we adapted from pure text retrieval, which we believe offers many advantages over more classical approaches. It presents the mathematical background and illustrates the method on a simple data set. It also reviews other basic models for retrieval.

In Chapter 6 we apply the material presented in Chapters 4 and 5 to the case of image+text documents. The integration of visual and semantic cues is presented in this part of the dissertation. This is probably the major contribution of our work. The chapter ends with an extensive evaluation of the performance of the method based on several increasingly large and complex data sets.

Another key issue we have addressed, namely user interaction, is the subject of Chapter 7. The exposition of the various interaction models is intermixed with result presentations based on the data and methods presented in Chapter 6.

The closing chapter 8 presents a discussion with the summary of the contributions and investigations. It also raises some questions that are open for future research. In a similar manner, each chapter is concluded with further research ideas and a summary of the achievements.

Chapter 2

Image retrieval systems: A systemic approach

Any system devoted to information storage, processing and retrieval must be designed with several important issues in mind.

- In today's information rich environment, where raw data is abundant, great care must be given to a scalable processing of raw data in order to produce meaningful information. Since *a priori* we do not know what use the data, and subsequently the information, will be put to, we must provide end-users and administrators with means to extrapolate novel processing techniques starting from established and well behaved existing information processing mechanisms. Keeping in mind today's large collections the scalability of any method should be considered of paramount importance during its design.

- The architecture must also allow a distributed deployment of the system, following the trend of network-based applications. The end-user should be allowed access to all possible data, information, algorithm and processing, within the limits of the network and computing resources linked to the user's work location.

- Finally any information-centered system must allow for sophisticated, but not necessarily user-visible, inter-media processing. The combination and interaction between the media-specific aspects of information (eg. the visual aspects of an image, or the perceptual aspects of audio) and those of its semantics (eg. the meaning of the depicted subject, or the meaning of a spoken document) must be taken into account.

All these aspects call for a flexible, open and evolvable architecture for the information retrieval system. We have analyzed existing approaches and designed a set of requirements we believe any such system should respect. Then, we have developed a software framework to allow the implementation and deployment of several multi-media information retrieval methods. This chapter describes this analysis and design in the following order: We present first (in Section 2.1) a study of existing systems from an architectural point of view. We then discuss in Section 2.2 the general complexity of an information system, its origin and how it can be managed. Section 2.3 then presents our proposal for a generic and flexible architecture for image retrieval. A summary of the results is presented in the closing Section 2.4.

2.1 Standard approaches to image retrieval

Image retrieval sprouted from the fertile soil of database research intermixed with a hint of information theory and document processing. The first research was aimed at text retrieval and passed through numerous phases, starting from abstract retrieval (medical, legal, etc.) to heterogeneous retrieval from the WWW of free-form complex documents. With the proliferation of media such as images, video and audio, and the steadily increasing availability of computing power and storage space, the angle of research started widening to accommodate these new types of content. New ingredients were poured into the brew of information retrieval, starting from signal processing, through artificial intelligence and bio-inspired techniques, up to more empirical and subjective issues of human-computer interaction and psycho-physics. All these disciplines find in information retrieval, a challenging area for endeavor.

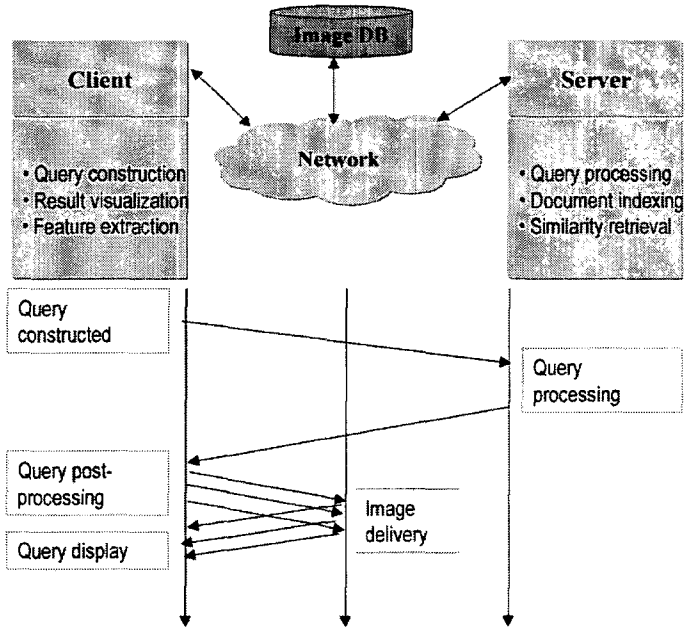


FIGURE 2.1: The simplest client server architecture for information retrieval.

An interesting fact (noted also in (Meghini *et al.*, 2001)) is that multimedia retrieval projects have been initiated by research groups in many disparate domains that seldom cooperate.

2.1.1 The early information retrieval systems

As a pioneering efforts in information retrieval, we can point to the Luhn (1957) where the basics were already exposed. Let us only cite Salton, Minsky, Winograd, Rijsbergen among the many researchers that have built the basis for information retrieval. For a survey of its history let us point to a few classical references: (Van"nobreakspace –"Rijsbergen, 1979) and (Baeza-Yates and Ribeiro-Neto, 1999). The scope of our work being that of integrated image and text retrieval, we will describe only those research efforts relevant to image media.

In the early 1990's the first work on content-based image, video and audio retrieval started. For image media the pioneering work was conducted at IBM's Almaden research center with the Query By Image Content project (QBIC) (Faloutsos *et al.*, 1994), which in a way became a standard reference for further research, both in terms of architecture and approach as well as for performance evaluation. From that point on, the research has undergone quite an explosion, papers from the computer vision, database and pure signal processing communities are published at an ever increasing rate. The problems encountered in the field are inherently difficult issues, ranging from image characterization (see (Rui *et al.*, 1999) for a survey) and understanding ((Crevier and Lepage, Aug 1997)), to image storage and network issues for data delivery ((Witten *et al.*, 1999)), security (see (Ling and Chang, 1998) for instance). More detailed reviews of these areas are given in the chapters concentrating on each aspect, we present here only the relevant material for system architecture.

A common oversight in many research efforts is that of the system architecture. Many scholars have developed high performance algorithms and strategies for dealing with single aspects of the information retrieval problem, rarely though we can see an all-encompassing effort to co-ordinate the various components of a system in a flexible and scalable way. Some notable exceptions include the work by Brunelli and Mich (2000) who propose a flexible enough architecture for several situations. Tao and Grosky (2000) also propose a complete architecture. But basically each group has developed its own system and many of those lead to the same basic principle: the general structure of an image retrieval system has converged to a client/server model. Figure 2.1 illustrates the simplest such configuration.

In this model, a centralized server manages the access to, and the retrieval of, the data. It accepts connections from remote clients, usually thin clients implementing just user interfaces and eventually the processing necessary to translate the queries to the adequate feature set. The network management part is usually based on HTTP or simple TCP/IP. Most implementations of the client interfaces are in

JAVA, thus taking advantage of the platform independence and browser integration. All the systems that fall into this category basically perform the same scenario, depicted on Figure 2.1:

1. The user specifies a query.
2. The client user interface pre-processes the query to a configuration compatible with the server.
3. The client sends the query over the network to the server, and awaits a reply.
4. The server processes the query, and retrieves a set of documents deemed relevant to the query.
5. The server sends the results over the network to the client.
6. The client processes the results and displays them to the user.
7. The user evaluates the results and either re-initiates the entire sequence or retrieves the documents that are relevant from the result list.

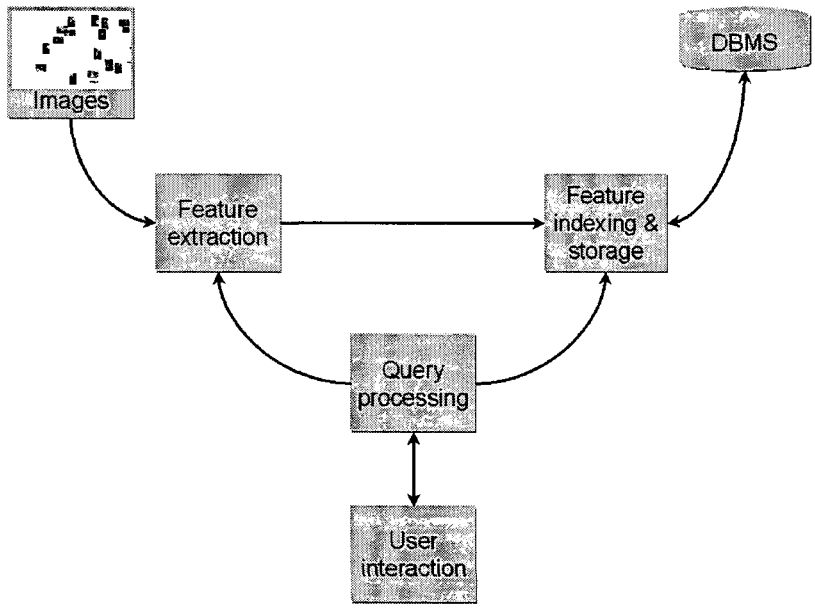


FIGURE 2.2: The data-flow through the major components of an information retrieval system.

Figure 2.2 illustrates the flow of data through the major components of an information retrieval system which are described below.

- The first and most widely studied component is the image analysis, or feature extraction stage. From any image, this component produces a set of numerical or semantic values which are associated with its content. The information produced by this stage is a sort of abstract of the data itself, or of its structure (segmentation of image regions or spatial relations), and must be flexible enough to permit any future use in the retrieval application. This processing is carried out during the database population phase, and can thus involve rather complex and lengthy operations which may also be performed under user supervision. We will return to this argument in Chapter 4.

- The second component is the indexing and storage component which is used to permanently store the image data and/or the extracted features. The essential aspect of this component should be its scalability to allow for large numbers of documents and speed to allow for a real-time querying possibilities. Usually this component is a database management system with some extensions for multi-media data and tailored indexing schemes. The topic of multi-media database systems is a subject of research in its own right, let us point the interested reader to the review (Yoshitaka and Ichikawa., 1999).

- The most visible part of the system is its user interface. This component must be designed in order to allow the maximum expressiveness to a user, to follow her/his task and adapt to it. Here again we will give more detail in Chapter 7.

- The core component of the retrieval system is of-course the query processing and retrieval engine. Based on the elements of a query, it searches through the stored features and index, retrieving references to the documents that are relevant to the query elements. More details follow in Chapter 6.

In (Smeulders *et al.*, 2000, p. 1372) we find a similar decomposition of the system into essential components:

> "As concerns system architecture, we maintain that a full-grown content-based retrieval system will result from the integration of a sensory and feature calculating part, a domain knowledge and interpretation module, an interaction and user interface module, and a storage and indexing module. For the system architectures [...], we conclude that most systems have an innovative emphasis understandably limited to one or two of these components. We feel there is a need for a framework for content-based image retrieval providing a more balanced view of the four constituent components."

And we must agree that a framework is needed in order to enhance the research communities' interaction and productivity. The roots of our solution are an open source framework alongside with an adapted and adaptive communication protocol for multi-media retrieval. More information about the protocol is given in Chapter 3.

2.1.2 A taxonomy of information retrieval architectures

In order to get a richer view of the existing information retrieval systems, we will attempt to evaluate the most common solutions in terms of a few key properties:

Distributed setup The system exploits the networked setup of modern computing facilities and allows access to its services from any remote computer. It also benefits from this setup to enhance the performance indices by parallelizing the processing in the essential components.

Flexibility The system has the ability to process new documents and document elements (image features, annotation, etc.) without the necessity to rebuild the system from scratch. It can answer queries of different types and add new query paradigms with minimum effort. Any addition to the system must leave previous applications in a functioning state. Migrating from one type of component to another (eg. changing the underlying database management system) must be as easy as possible. In other words, the system is able to grow in complexity with as few modifications as possible.

Scalability The same system applied to two sets of data of very different size suffers at most a proportional decrease of efficiency and above all effectiveness[1].

Completeness The system is complete in the sense that all major user tasks in an information retrieval environment are provided with an adequate solution. The completeness can be evaluated only in relation to a set of defined tasks. On the other hand the system will not ignore any useful data that might eventually become available for any documents (annotation, user profiles, new description methods).

Some of the most well-known information retrieval systems were compared in terms of architecture properties. The general remark we can make is the rather rigid architecture of most of them thus compromising the flexibility. Furthermore most of the analyzed systems don't exploit all the side information in the best manner. Usually, the semantics is not well integrated with the visual aspects at least from the query point of view. A synthesis of the findings is given in Table 2.1, we also point to (Veltkamp and Tanase, 2000) for a more detailed description of 39 most well known systems.

[1] Efficiency is the term usually employed to describe the execution time, disk I/O, and memory requirements. Effectiveness is used to characterize the property of a system to satisfy the user expectations in terms of result relevance.

Table 2.1: The taxonomy for system architectures.

System	D	F	S	C	Description
QBIC (Faloutsos *et al.*, 1994)	yes	+/-	yes	no	QBIC allows for updating of the document set and can be exploited in a distributed fashion, it lacks some essential query paradigms and doesn't exploit all side information.
Virage (Virage Inc., 1998)	yes	yes	yes	no	Being an extension to a database management system, the Virage system is fairly flexible, any effort to enhance its functionality is guaranteed to leave previous applications in a consistent state, as QBIC, it lacks support for some query paradigms and doesn't exploit side-information automatically.
ImageRover (Sclaroff *et al.*, 1997)	yes	yes	yes	no	The system is based on a modular architecture, where analysis rovers and robots are dispatched on the network to index the WWW. The only drawback in this taxonomy is the lack of support for new components and the quite poor support for side-information.
MARS (Porkaew *et al.*, 1999)	yes	no	?	no	A distributed but rather fixed system, lacking of the necessary flexibility.
Viper (Marchand-Maillet, 1997—)	yes	yes	yes	yes	The Viper system is one of the most flexible and complete systems. It too is based on the Multimedia Retrieval Markup Language (MRML) and communication protocol (see Chapter 3) and offers many query paradigms.
CIRCUS	yes	yes	yes	yes	The CIRCUS system was designed exactly with the four properties in mind. The only problem that may be encountered is in terms of database updating, at least in its principal method: Latent Semantic Indexing (see Chapter 5 and Chapter 6).
D: Distributed, F: Flexible, S: Scalable, C: Complete.					

2.2 Complex versus Complicated systems

As a preliminary step before presenting our system in more detail, the essential argument must be made about system complexity. While complex and complicated are often used as synonyms, they are subtly different adjectives. Complex applies to systems that are inherently difficult, where the simple components intricately connected with simple relationships, behave in a way more difficult to fathom and describe. Complicated is a participle, meaning that some external influence has modified the system's behavior in order to render it less obvious to understand or describe.

We can argue that any system involving user subjectivity and specific tasks with dynamic goals, becomes complicated. The user, be the system complex or simple to start out with, renders the system complicated and increases the complexity by a non trivial amount. In other words, the difficulties and problems arising in image retrieval can almost always be traced down to the human component. These can arise at both the semantic and syntactic levels of interpretation. At the semantic level the subjective interpretation of relevance of a query result is dependent on the user's socio-cultural context or present task. At a syntactic level the perceptual system of one user differs from that of another. It is easy to see that a system devoted to answering only a restricted set of queries on a set of images restricted to a fixed domain, can be designed to be very effective. Unrestricted image types and, even more, unrestricted query types can produce complexity which is as yet intractable.

The CIRCUS architecture presented in Section 2.3, can produce image retrieval systems that are quite complex while being based on simple components. We will try to give a notion of system evaluation that will take the complicating human factor into account.

2.3 The Content-based Image Retrieval and Consultation User-centered System: CIRCUS

The project entitled *Content-based Image retrieval and Consultation User-centered System* (CIRCUS) is an effort to develop a framework for image retrieval of distributed, heterogeneous, annotated image collections. A large part of CIRCUS was designed and implemented in collaboration with Prof. Thierry Pun's Vision and Multimedia Lab at the University of Geneva, especially David Squire and Wolfgang Müller. The whole idea came about by our joint wish to achieve compatible system implementations, for exchanging raw data, results, evaluations, user profiles and any by-product of experimentation. The major result of this collaboration is the MRML communication protocol (see Chapter 3). Extending the basic client-server idea to multiple components that are running concurrently in a distributed setup, resulted in the core ideas for the CIRCUS system.

CIRCUS is constructed on top of an extension to a client/server architecture with an open and formally defined communication protocol MRML (Müller *et al.*, 2000*a*) which is available under the GNU Public License (Maillet *et al.*, 2000—). This architecture allows for various user interfaces to connect to a set of retrieval servers implementing different methods or operating on different collections. It also provides means for automatic benchmarking and minimal effort meta-search engine construction. A series of user interaction paradigms, based on interactive visualizations and summarizing abilities are presented in detail in Chapter 7. Figure 2.3 illustrates this architecture. The following section will describe each entity more closely.

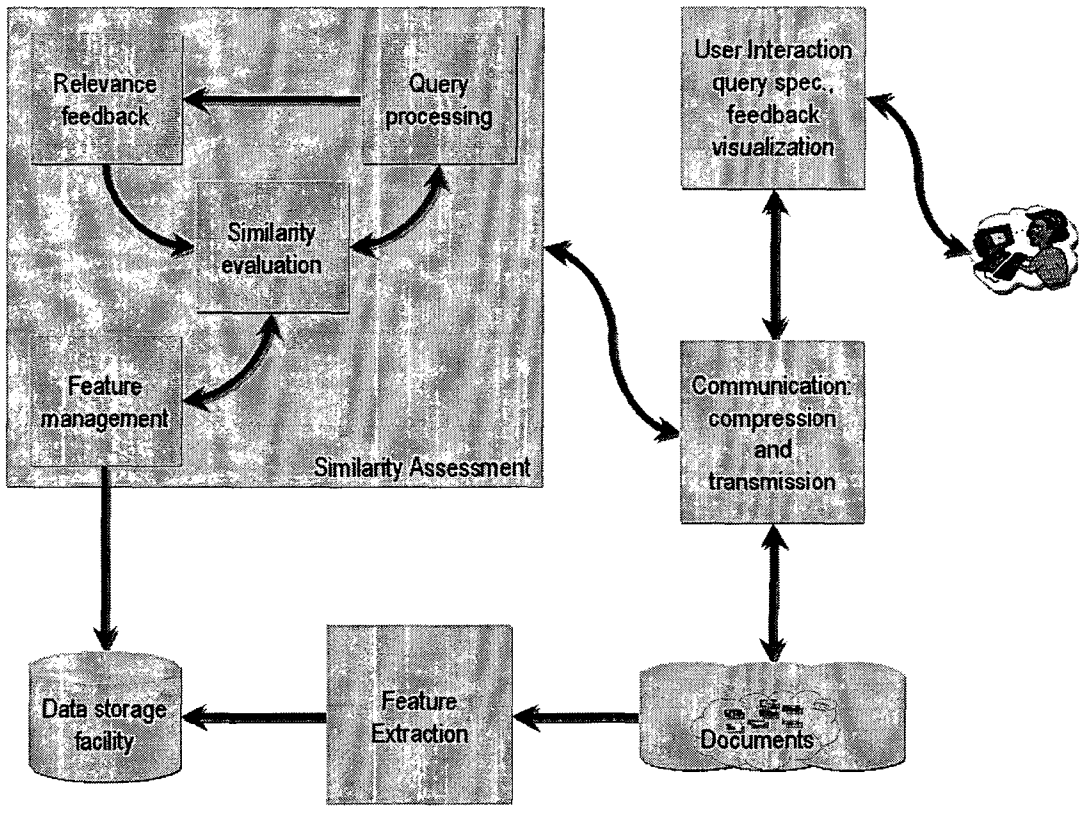


FIGURE 2.3: The components of the CIRCUS System.

2.3.1 The CIRCUS components

CIRCUS consists of a network of inter-operating components that communicate using either network facilities, permanently stored files or operating system IPC methods[2]. Any process is a sequence of messages initiated by the user and propagating among the various entities in the system. On each machine a group of correlated components is running to provide a larger functionality. Typically a user client application, a server acting as retrieval engine, and possibly a set of related servers accessed either directly or through a multiplexing component. The diverse components can be subdivided into eight categories:

[2]Inter Process Communication is usually achieved through pipes on UNIX-flavored systems or shared memory.

Control layer: communication Each component of CIRCUS is associated with a communication component that acts as a control and validation layer. Thus each component can interact with other components by exchanging messages in a dedicated XML format[3]: Multi-media Retrieval Markup Language (MRML). The basic structure of the message is an XML tree containing data of either a query for information (relevant documents, document properties, document constituent elements, etc.) or a response to such a query. So for instance the user interface's communication component can interact with a meta-server component's. The latter will extract from the message tree the portions that are target to it alone and forward the rest to a set of pure server components as if it had been a user interface component. The results from the individual replies to the query are then assembled in a single representation and sent directly to the user interface component.

This is the *glue* component of the system in the sense that it sequences and regulates the data exchange between the different components. We refer to Chapter 3 for a more detailed description.

Data gathering The first step to structuring a collection of documents is to locate the raw data and either down-load it to where it will be pre-processed for analysis, feature extraction and so on, or upload the processing elements to the data storage and let the processing take place on the data server. This role is played by the data gathering components. Currently, the only data gathering component available is one that down-loads a set of documents for processing, possibly by following recursively an XML-like structure of document references and invoking the document analysis components. Had the documents been managed by a database management system (DBMS), the component could be replaced conveniently by a procedure in the DBMS that would invoke the document analysis component on each new or modified document.

Document analysis The role of these components is to extract from a document or set of documents the characteristics that will best describe it, allowing comparisons to be made involving similarity and discrimination. The type of processing involved is not necessarily known in advance, and thus each component must describe the features it generates in a XML-like structure giving semantics to the created information. Along-side the document analysis component we must also have a sibling component that acts as assessor of feature comparisons. These components are implemented in more efficient ways if the nature of the features is fixed once and for all, but that is precisely the kind of situation we aim to avoid, hence our choice of a heavier but more flexible solution.

Once extracted, the features are passed on to the components that are to create a retrieval strategy from them and prepare them for storage and indexing. A set of such components is described in Chapter 4.

Feature management These components manipulate the features into a form most suitable to fulfill the requirements of the query processing components. They implement the model of retrieval.

- The most widely spread model is the vector space model (covered synthetically in (Dominich, 2000)), where each document is represented by a vector of numerical values, thus becoming a point in a high-dimensional space.
- The inverted file model (Brown *et al.*, 1994) associates to each feature a list of documents containing the feature and thus allows for fast processing of feature based queries.
- Probabilistic models (Vasconcelos, 2001) associate to each document and feature the occurrence probabilities, the conditional probabilities and the marginal probabilities. The queries are then represented as a maximum likelihood search Bayesian process.
- Other hybrid models can also be implemented and the principal model we are proposing in this work is the Latent Semantic Indexing Model in Chapter 5.

More detail on the retrieval modes is available in Chapter 5 along with a taxonomy of their capabilities.

The stored features are thus indexed according to the model and access to them goes through a peer component. The tight integration of feature management and query processing, all goes through the use of the MRML messaging format.

Query processing The query processing components translate the query into the features that characterize it according to the requested model and method, using the document analysis components

[3]eXtendible Markup Language (XML) is a method of data and document structuring according to formally specified grammars and formatting rules.

and the feature management components. It then retrieves, through the model representation and method, the references to documents which have high relevance to the query. Here again tight coupling is assured by the use of MRML. Up-stream in the process flow, the user interface component or batch processing component can steer the query processing by adaptive parameters in order to fine-tune its performance. Different query processing components are described in chapters Chapter 6 and Chapter 7.

User interface The most visible part of the entire system, the user interface component, is the entry point to the system. These components present the user with meaningful and intuitive ways of formulating queries and, display results in ways that stimulate the user and enhance performance. Chapter 7 is entirely dedicated to this subject.

Batch processing These components are designed to operate along a user defined sequence of queries, or as "drivers" for, performance evaluation, statistics gathering, and maintenance work. They interact with the query processing components in the same way a user interface component would. They are just a convenience functionality, but priceless when it comes down to evaluation, benchmarking and model comparisons.

Meta-server The meta server components act as glue between user interface components or batch processing components and a multitude of pure server components like query processing components. They can fan-out a query to a set of servers and ensure the convergence of the results to the initiator or can act as multiplexers, gathering the results and merging them into a single MRML message to ship back to the initiator. These components are useful for the integration of several retrieval methods or for unifying different data sources (distributed document collections).

2.3.2 CIRCUS Implementation

Most of the components described in Section 2.3.1 were implemented. Others were just described and formalized. The prototype system was written either in pure JAVA or in JAVA and C/C++/FORTRAN90 through JNI[4]. This approach allows for computationally intensive code to be written in the most efficient manner, while using a single implementation of the communication and control framework. Machine specific code does not exist, the external libraries that are accessed through JNI, use nothing but standard mathematic libraries and file I/O. Porting to a different target system should thus not be a problem.

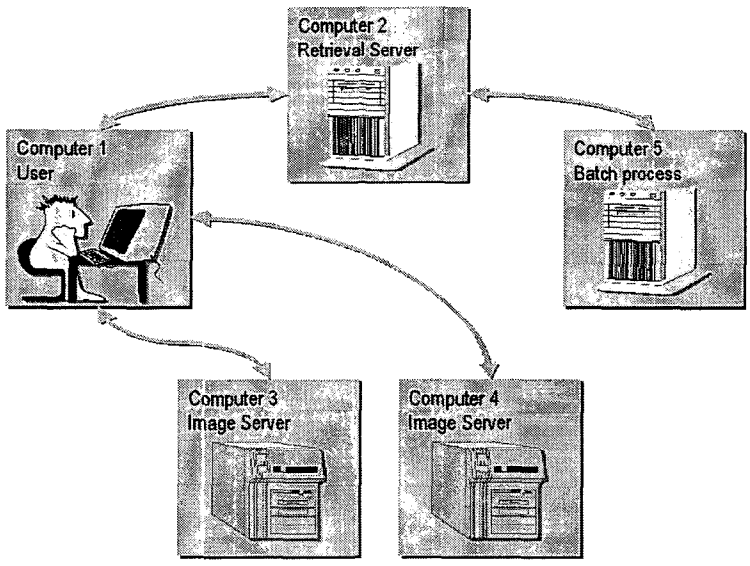


FIGURE 2.4: A sample setup of the CIRCUS system in a distributed computing environment.

Figure 2.4 depicts a sample deployment of the CIRCUS system. We see several computers running different components and performing different tasks. The user of the system is working at `computer1`,

[4]Java Native Interface, allows linking JAVA code to external libraries compiled from any language, bringing higher execution performance to JAVA applications.

the query is being processed on `computer2` and data is being retrieved from `computer3` and `computer4`. At the same time `computer5` is performing maintenance and evaluation from a batch processing component.

2.3.3 Advantages of the CIRCUS architecture

Since its first days, image retrieval has been more of an art form rather than a strict, hard, scientific endeavor. Blending and mixing the right ingredients seems to be the key to an efficient and effective system. In this cooking game, we believe that much can be gained from using interchangeable ingredients; replacing a given processing with one that is compatible but based on entirely different models. This allows us to move quickly from an idea to a functioning implementation, then to experimenting and finally to performance evaluation. So, the major goal in designing CIRCUS was to allow for maximum inter-operability.

The communicating components architecture meets the flexibility requirements very well, it decomposes the problem solution into small, easy and controllable sub-problem solutions. The implementation times are reduced, and reusability quickly pays off the slightly longer design process, and sub-optimal implementation.

A second, seemingly counter-intuitive, advantage to an architecture such as CIRCUS is its potentially higher efficiency. This potential is due to the distributed architecture and to components that can act as scatter/gather elements in a concurrent or parallel application design. For instance, document analysis can be performed in parallel, as can some retrieval processes, and going from a sequential to parallel processing is a matter of instantiating more components and adding a meta-server controller to scatter/gather the results.

2.4 Summary and discussion

In this chapter, following a presentation of the architectural characteristics of current solutions for image retrieval, we proposed a flexible and scalable solution. The scope of the presented architecture is beyond just image retrieval, it can easily be applied for multimedia retrieval systems. The basic model is that of communicating components that exchange processed data and processing descriptions through a uniform protocol and message format based on XML and described later in Chapter 3.

We also presented a framework system: Content-based Image Retrieval and Consultation User-centered System, or CIRCUS. The motivation behind this implementation is the desire to test different strategies for performing the same basic tasks, with minimal overhead for development. The rich set of tools offered by the CIRCUS implementation will be studied in the following chapters according to their application areas.

We finally presented a mathematical framework for the quantitative and qualitative evaluation of the produced systems, by introducing basic performance evaluation criteria like precision, recall, efficiency and user satisfaction.

The still open questions raised in this discussion, and good thought experiments for future research, are mainly:

- Analysis of media-specific components and their requirements.
- Control framework specification for managing the interaction among the processing blocks.
- Acquisition and validation of ground-truth data for performance evaluation.

Chapter 3

The Communication layer: Unified data sources and inter-operable components

As exposed in Chapter 2, an essential part of a flexible and distributed image retrieval system is the communication protocol. Any application involving a remote access to information must be designed with a robust and extensible communication layer. The semantics of the exchanged information, queries or results, must be contained in both the abstraction layer and the actual information. Several independent components of the system must be able to interact knowing the meaning of the interaction. The end-user, or even more so the system administrator, must be able to understand this exchange of information and interpret it unambiguously. These considerations motivate a careful analysis of the multimedia retrieval tasks and lead to the specification of a versatile protocol described in detail in this chapter. A large part of the design and implementation of this protocol was carried out by the Vision and Multimedia Lab of the University of Geneva. Especially in collaboration with David Squire and Wolfgang Müller.

In Section 3.1, we first present some previous efforts to formalize and make more uniform the access to multimedia data and the retrieval task. In Section 3.2, we define a uniform, transparent and unambiguous access to multiple multimedia retrieval services. The extension of uniform access, in terms of user interaction, is exposed in Section 3.3. Then a general description and specification of the multimedia retrieval markup language (MRML) is presented in Section 3.4. A final discussion with future directions of investigation is proposed in Section 3.6. Appendix 3.A provides additional detail and a complete reference to the MRML protocol.

3.1 Multi-media information access protocols

Together with the increase of the volume of information available through the Internet, we have witnessed a rapid increase in the amount of different media available to users. Starting from the simple textual and numerical data, the information has evolved to illustrated documents, to audio, video and intermixed interactive presentations. The bare number of formats in which the actual data is stored is overwhelming. The advent of markup-languages for structuring the data and uniformizing its description, has spurred a range of research efforts.

The ideas and dreams of interlinked documents date back to the beginning of the century, like in Bush (1945),a *pseudo* sci-fi article *As we may think.* The first popular implementation is the Hypercard system of Apple Inc. from 1987. In 1989, extending these ideas to the networked computer environment, Tim Berners-Lee and Robert Caillau, at CERN, developed HTML and the associated HTTP. The goal was to allow different users on different computer systems to access several information sources in a uniform way. The idea of linking documents through document parts and having a software application allowing to retrieve the documents based on URIs[1] was going to revolutionize modern information technology. The simple way of linking documents was enough to start out with.

Eventually, this inherently manual process, mainly performed at the time of document creation, and the huge number of documents available, has rendered the pure hyper-media access too cumbersome.

[1] Uniform resource identifier.

Furthermore, there has been scarce advancement in the domain of access by the contents of the non-textual media.

Compression, delivery, quality and reliability have all been addressed and though no definite solution has yet been agreed upon, the efforts of the standardization authorities are becoming effective. Table 3.1 illustrates some of the major achievements in the previously mentioned areas. The issue of content-based access and the associated semantics have only recently become the center of interest (eg. in the MPEG 7 standards ISO/IEC (2002)). Although these attempts do offer a valid description and semantics to single documents, they do not cover the ramifications in the case of an interactive task, or of the user accessing document collections from a remote service. This is the scope of our investigation on MRML.

Table 3.1: The most popular communication protocols for information retrieval, transmission and compression.

Name and References	Description
Internet Imaging Protocol (IIP) I^{3}A (2002)	This is a multi-resolution, tiled and annotated image storing and transmission protocol that allows for access to sub-regions of an image at a given resolution.
HyperText Transfer Protocol (HTTP) W3 Consortium (2002*a*)	Among the simplest protocols for requesting resources from a distant server. Simple means of passing parameters without their semantics is provided with the Common Gateway Interface (CGI). The protocol provides for transferring either a file with associated type, length and dates to the client application, or transferring the result of an executed CGI script as if it were a file.
MPEG-7 ISO/IEC (2002)	This is an ongoing effort to represent multi-media documents in an efficient manner, through high grade compression, embedded encoding and synchronization. It also provides for the content description using a syntax based on XML Schema[2].
SMIL W3 Consortium (2002*b*)	This HTML-like document description language, using media of diverse data formats and associated protocols (based on UDP usually), is dedicated to the transmission of streamed and synchronized media like video, audio, or presentation like multimedia documents. They offer basic navigation through a single document and no capabilities for retrieval from collections of documents.

From a more abstract point of view, hints and insights relevant to retrieval protocols can be seen in works by Chang et al. Chang *et al.* (1999). In their paper, we find a taxonomy of query tasks based on user tasks. This extensive taxonomy leads the authors to suggest a uniform access framework for retrieval services.

3.2 Defining uniform access to image retrieval services

In a distributed environment, the answers to user's information needs may come from multiple data repositories (WWW sites, databases, data warehouses, etc.). The user cannot know in advance where the answers lie, since otherwise there would not be any problem to solve. The user has to access all these data services and query whether relevant material is present. She/he must learn to interact with all the potentially very different retrieval systems. For example one system might use a text-based interface, another a query by sketch interface, yet another a query by example interface. Even more problematic is the fact that the user must also be aware of the *existence* of the numerous retrieval systems. To make an analogy in a country where multiple telephone companies cater to the population, it is desirable that a centralized company maintain the phone-books with all the numbers of the subscribers. Otherwise, the poor user would have to page through all possible phone-books in order to find the number of the correspondent. Likewise, we can hypothesize a future where data exchange will not be limited by so many commercial and legal implications as today, and where the different information caterers will gladly cooperate to unify the access to their data.

[2]XML Schema is the successor of the Document Type Description, the standard way of specifying the grammar of a well-formed XML document. It is itself a well-formed XML document and so can be analyzed with any XML parser.

On the development side of the systems, there is an unnecessary waste of resources since most parts must be coded from scratch. From the researchers point of view, the lack of a uniform and standard framework is a huge disadvantage. The time span of a PhD or a normal research project devoted to image, or video retrieval, will typically not allow a single person, or even a small group, to finish a complete system. Much time and effort will be wasted on basically repeating the work of building a server and client application. A considerable part of that time would be devoted to the communication between the two afore mentioned parts. Instead of concentrating the energy on new research areas, like a new image texture characterization, it is wasted on implementation issues.

Further, the domain of multimedia retrieval still lacks reliable benchmarks and evaluation systems. One reason for this is the lack of good data to perform the tests with, the other reason is that no two systems can be automatically evaluated with the same evaluation software.

In any of these cases a simple solution exists. Alas what is simple is not always the most profitable. The basic idea is that any retrieval service must be able to communicate with its client using a fixed, yet extensible protocol and message format. So analyzing the retrieval task, and implementing a protocol and message format covering the tasks of retrieval is the goal of our investigation.

3.3 Multi-media retrieval task analysis

We present here a brief analysis of typical retrieval tasks in a multi-media environment. A more detailed analysis is given in Section 7.2 and Appendix 7.A. We concentrate here on the messaging between a client application and the service providing application.

We have identified five basic task elements that any complete retrieval system should provide. They are described in some detail below.

Capabilities negotiation As not all servers and clients have all the described capabilities, the two parties must be able to exchange the capabilities information. Any capability that cannot be fulfilled by the replying party will be ignored.

Document access and navigation The user of the system wants to access a known set of documents, identified by a list of URIs. The client, after having received from the service the list of URIs, is responsible for the request of the document contents based on the URIs. The server must be capable of returning the content either in its entirety or only as parts of the requested document (image region, audio/video sub-frame, chapter or paragraph).

Query specification and interpretation The client must be able to provide at least basic query specifications including textual elements, visual elements like examples either from the collection or not and media-specific elements like color properties for images and video or pitch and envelope for audio recordings. The client and server must be able to negotiate the interpretation of any query submitted to the server. This includes but is not limited to the type of the query elements (annotation, visual features, audio features, meta-data).

Collection overviews and browsing The server must be able to provide a structured overview of the collections it is capable of searching. Typically, if the documents have been categorized, the server should be able to produce a summary of the categories and sample contents for each category. The client should be able to interpret these unambiguously and present them to the user in whatever form it deems most appropriate.

Query process management Both the server and the client must be able to consider the query process as an evolving interaction. The client should keep some track of history and the server should know whether a specific query is dependent of the previous queries in a transaction-like fashion, or a new interaction. If the server is capable of mid-term or long-term learning, it must be able to identify the user and keep track of the changing parameters of the query for future interaction.

3.4 Multi-media retrieval markup language

The previous analysis has lead to the design of the multi-media retrieval markup language (MRML). This section describes the protocol and format of messages in some detail. A formal specification is available through the web-site `http://www.mrml.net` and discussed in Appendix 3.A.

3.4.1 MRML overview

MRML is a living standard, meaning that it is as yet not finalized and will not be for some time. In Section 3.4.2 we give some more ideas as to why this should be the case. It is an XML based message format associated to a simple state-machine protocol. Its main goal is to specify the information exchange between servers and interactive clients, but as will be seen, it has a wider application range. Figure 3.1 illustrates a simple setup with multiple clients and multiple servers. It places MRML in a higher level of abstraction than the server dependent access to the documents which can happen through a dedicated DBMS.

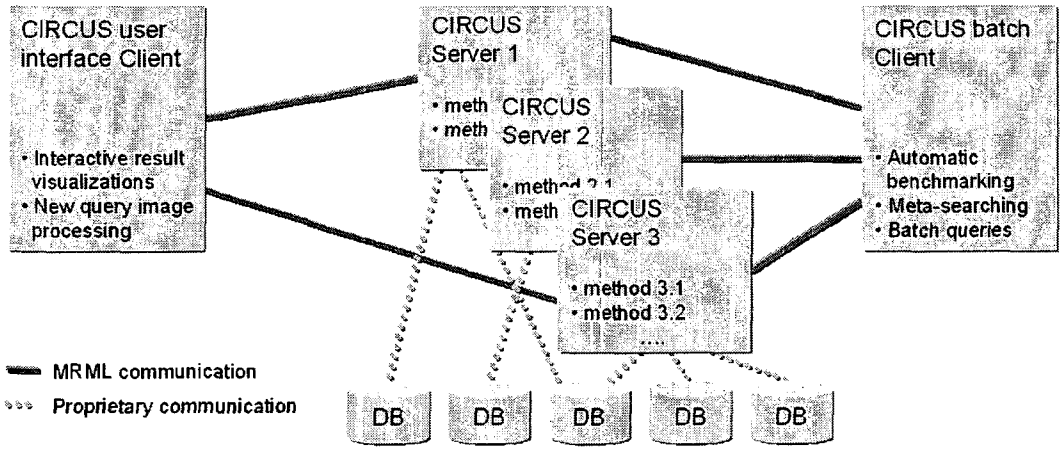


FIGURE 3.1: The setup in which MRML operates. Several different client applications can access several different servers that all process queries in particular ways. This communication is managed by MRML. Non-interactive clients (like benchmarking for instance) use the same interface to the retrieval service.

The benefits of an open protocol and message format can be understood by considering the success of standards like SQL or ODBC[3]. Why not use these standards then? Simply because while SQL is well suited for alpha-numeric database queries, it offers as yet little support for multi-media databases and the ever evolving Internet based distributed applications. It is also a relatively cumbersome framework.

We have decided to use an XML based approach mainly because of the long term support that XML is expected to receive and of the availability of free parsing and evaluation software. More complete systems like CORBA[4] incur a high complexity and do not provide enough flexibility. XML is becoming the standard data exchange format in industry, business and research. It also has the advantage that the data remains in a human-readable format which in terms of research and development is a welcome feature.

The transport protocol is simple TCP/IP or eventually HTTP-tunneling. HTTP-tunneling is a means to provide access to different services than the standard web-servers if firewalls are restricting access to either party. It consists of a web-server extension like a JAVA servelt, or CGI script that receives the request through HTTP, passes it on to the actual server, than waits for the reply and sends it back to the client again over HTTP. From the firewall point of view the only traffic is HTTP, even though the contents of the messages and the protocol are MRML.

As conceived, MRML facilitates a bottom-up development approach, by separating the communication issues from the research of an ideal query language for multi-media retrieval. The key feature is its extensible nature, where any kind of query can be accommodated without compromising the standard. Guidelines for extending the protocol are given in the detailed specification.

MRML also supports user interfaces that are automatically configured through the use of property-sheets. Each algorithm and query paradigm have their own parameters with associated semantics. The server that may offer a set of algorithms for retrieving from a given collection can also offer for each algorithm a description of the parameters, their types and ranges. The interface constructs a visual display of these parameters on the fly.

[3]Structured Query Language and Open Database Connectivity are two cooperating standards that unify the access to Database management systems form a wide range of applications.

[4]Common Object Request Broker Architecture is a standard architecture for designing distributed systems. It has numerous advantages but the complexity involved is usually too high a price to pay in an already heavy environment of multi-media retrieval.

3.4.2 MRML design goals

When designing MRML, the goals we have put at the top of our priorities were all targeted at the research community, but most of them are beneficial to any industrial and business application. We have put much effort in ensuring that MRML has the following features:

Simplicity MRML was designed to minimize the development costs of systems that wish to be compliant. It has a simple well-formed XML grammar, but it is not necessarily accompanied with a document type definition (DTD), thus allowing us the use of non validating parsers, and even decreasing execution overhead.

Inter-operability Any compliant application can communicate with any other application, thus enabling several interesting scenarios:

1. Meta-search engines can be built from applications that behave both like servers and like clients. Routing the initial query to a set of destination search services they combine the results into a single reply sent back to the user.
2. System comparison and evaluation software will be able to access different systems and compare them on even grounds.
3. Any user accustomed to a given interface need not learn a new interface, but can access any multi-media retrieval system that complies to MRML, from his favorite one.

Extensibility Any team wishing to extend the capabilities of their system can do so without prior permission from a standardization organization. The only requirement is that the team publish these extensions on the commonly administrated web site `www.mrml.net`. As the life-cycle of MRML goes on, the extensions can become part of the newer versions of MRML.

Graceful degradation Related to the extensibility, we stipulate that any server or client unable of dealing with a certain XML element of the query or reply will simply ignore it and its contents. It can eventually notify the other party of the fact. This guarantees backwards compatibility but does not hinder the development of new extensions. Simple rules of XML naming should be applied and any extension documented and stored in the MRML repository.

Interaction data exchange Teams that are interested in collaboration, can exchange precious data on user relevance judgments, qualitative impressions and performance evaluations. This is achieved by the common message format which can be logged and exchanged as is. Any compliant application will know how to interpret them.

3.4.3 MRML message format

Each MRML message is a well-formed XML document. A sample such message is given on Figure 3.2. In all further examples the standard XML preamble (first two lines) is stripped. Each MRML document exchanged can contain a set of different MRML messages in a single transmission. In future implementations where permanent connected communication will be specified for MRML, the stream of messages will eventually be enclosed in a single MRML document.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE mrml SYSTEM "http://www.mrml.net/mrml.dtd">
<mrml>
message_1
message_2
...
message_n
</mrml>
```

FIGURE 3.2: The basic structure of an MRML message.

The current version of MRML provides 18 predefined message types. Any extensions should be carefully designed to keep these types functioning and to conserve their semantics. We give a brief description of each message. Their sequencing is the subject of Section 3.4.4.

<begin-transaction/>. If the message contains a <begin-transaction/> element it also must contain a <end-transaction/> element. The intervening elements constitute the object of the transaction. It simply instructs the server to keep trace of the actions of the client and to implement basic atomicity principles to the interaction.

<end-transaction/>. See <begin-transaction/>. This message ends a transaction.

<get-configuration/> is the message that requests from the correspondent a list of capabilities delivered in a reply of type <configuration-description/>.

<configuration-description/> is the reply to the <get-configuration/> message. It contains all the message types the party accepts and may send. It also contains a list of message types that are extensions of MRML with a human-readable description of the extension. The way the answer is interpreted is part of the extension mechanism.

<get-session/> is a request from a client to the server following the reception of the configuration. It means that the server should send back the list of available sessions to the user identified as part of the <get-sessions/> message.

<session-list/> is the reply to the <get-sessions/> message and contains a list of session identifiers and names the server has stored for a given user.

<open-session/> is the request to the server to open a given session.

<close-session/> is sent in order to end a session.

<delete-session/> is sent when the user has decided that the session should not be kept permanent. The sessions may also be deleted by the server without explicit request from the user.

<rename-session/> changes the name of the specified session to a new one.

<get-collections/> requests from the server a list of document collections it can retrieve from.

<collection-list/> is the reply of the server to a <get-collections/> message. It consists of a list of collections identifiers and names and associated <query-paradigms/> (see Appendix 3.A.2 for more details).

<get-algorithms/> is sent to the server to find out which retrieval methods are available for a given collection and given query paradigm.

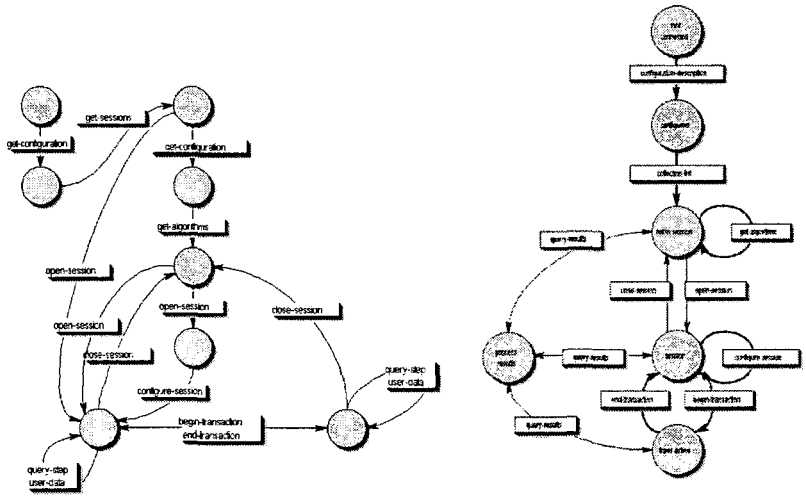
<algorithm-list/> is the list of available algorithms for a given collection and query paradigm. It can contain for each algorithm a <property-sheet/> that will define the algorithm's parameters and allow the interface to configure the display.

<query-step/> is the basic query message that is either a new query or the continuation of an interactive session. It contains various elements according to the type of query and query paradigm. See Section 3.A.2 for more details.

<query-result/> is the basic result message for a query step. It can contain a list of document URIs and navigation information, as well a certain amount of display related information if requested by the query step.

<user-data/> is the message a user interface sends back to the server for interactive sessions. It is used to track the query process and is usually a history browsing trace of the session. It can contain any other useful information the user is willing to submit.

<error/> is the message that can be sent alongside or in place of any other message. It signals anomalous behavior. It contains the level of seriousness and a text message along with a code among the predefined codes of the specification.



(a) Server (b) Client

FIGURE 3.3: The state machines of MRML parties

3.4.4 MRML protocol

The above described messages are exchanged according to a fixed sequence. MRML communication is similar to a Remote Procedure Call (RPC), meaning that there is no permanent connection between the server and the client. The server and the client exchange the information following state machines illustrated on Figure 3.3(a) and Figure 3.3(b) for the server and the client respectively.

The client and server go through an initial handshaking phase where configuration information is exchanged and capabilities of the communicating parties are made known. The client initiates a sequence of queries either grouped in sessions, if the queries are of an iterative nature, or in absolute single step queries. This querying phase can be interleaved with additional session management, transaction management and capability enquiries by the client. Ideally the user also closes each session before opening a new one, but this is done implicitly if necessary. Table 3.2 illustrates a sequence diagram for a simple situation.

CLIENT		SERVER
get-configuration	→	
	←	configuration-description
get-sessions	→	
	←	session-list
open-session	→	
	←	acknowledge-session-op
get-collections	→	
	←	collections-list
get-algorithms open-session	→	
	←	algorithm-list acknowledge-session-op
configure-session query-step	→	
	←	query-result
query-step	→	
	←	query-result

TABLE 3.2: The sequence diagram of a simple interaction between a new client and server. The client connects and handshakes with the server, than initiates a session and a series of query steps.

3.5 MRML exchange examples

To clarify the above descriptions we present here first several MRML messages from the hypothetical scenario of Figure 3.2 that give a feel of the MRML functionality. Then we show some examples of applications of MRML in the implementations of CIRCUS and of the VIPER systems.

The initial handshake results in the client having a list of collections the server is capable of retrieving from. The messages exchanged can be seen on Figure 3.4.

Client	Server

`<mrml>`
`  <get-configuration />`
`<mrml>`
	`<mrml>`
	`  <configuration-description>`
	`  <supported-message name="query-step" />`
	`  <supported-message name="open-session" />`
	`  ...`
	`  <extension-message type=element direction=receipt`
	`  name="CIRCUS-get-overview">`
	`  <message-description>`
	`  This message request the delivery of an overview of`
	`  a collection using the specified overview algorithm.`
	`  </message-description>`
	`  <message-DTD>`
	`  <!ELEMENT CIRCUS-get-overview EMPTY>`
	`  <!ATTLIST CIRCUS-get-overview`
	`  collection-id CDATA #REQUIRED`
	`  overview-algorithm-id CDATA #REQUIRED>`
	`  </message-DTD>`
	`  </extension-message>`
	`  </configuration-description>`
	`</mrml>`

`<mrml>`
`  <get-collections/>`
`</mrml>`
	`<mrml>`
	`  <collection-list>`
	`  <collection collection-id=1`
	`  collection-name="COREL CDs">`
	`  <query-paradigm-list>`
	`  <query-paradigm query-paradigm-id=1 query-pradigm-name="QbE"/>`
	`  <query-paradigm query-paradigm-id=2 query-paradigm-name="Browsing"/>`
	`  <query-paradigm query-paradigm-id=3 query-paradigm-name="Keywords"/>`
	`  <query-paradigm query-paradigm-id=4 query-paradigm-name="Color"/>`
	`  <query-paradigm query-paradigm-id=5 query-paradigm-name="Sketch"/>`
	`  <query-paradigm query-paradigm-id=11 query-paradigm-name="Text+Visual"/>`
	`  </query-paradigm-list>`
	`  </collection>`
	`  <collection collection-id=2`
	`  collection-name="Corbis">`
	`  ...`
	`  </collection>`
	`  </collection-list>`
	`</mrml>`

FIGURE 3.4: The initial handshake between client and server (from scenario of Figure 3.2).

For a given collection the user requests the available algorithms and issues an `<open-session/>` command. The list of available algorithms and their configuration is returned by the server, along with a reply that a new session is opened. Until a new session is opened or the current one closed all messages will carry the session identifier. This exchange is depicted on Figure 3.5.

Client	Server

`<mrml>`
`  <get-algorithms collection-id=1`
`  query-paradigm-id=11 />`
`  <open-session user-name="demo"`
`  session-name="horses" />`
`</mrml>`
	`<mrml>`
	`  <algorithm-list>`
	`  <algorithm algorithm-id=12 collection-id=1`
	`  algorithm-name="Latent Semantic Indexing (opt.7)">`
	`  <property-sheet property-sheet-id=156 property-sheet-type=panel`
	`  caption="Latent Semantic Indexing (opt.7) settings">`
	`  <property-sheet property-sheet-id=157`
	`  ...`
	`  </property-sheet>`
	`  </algorithm>`
	`  <algorithm algorithm-id=1 algorithm-name="Fast WP KLT." collection-id=1 />`
	`  </algorithm-list>`
	
	`  <acknowledge-session-op session-id=173`
	`  session-name="horses" />`
	`</mrml>`

FIGURE 3.5: The client retrieves the set of algorithms that correspond to a given collection of documents and simultaneously opens a new named session (from scenario of Figure 3.2).

Figure 3.6 presents a typical query step. First though, the user configures the session to use algorithm *Latent Semantic Indexing (opt.7)*. The query step consists of a keyword search with keyword "horse". Finally the server replies with a list of results. It also sends an extension message of overview information only relevant to the CIRCUS user interface.

The previous examples are a sample exchange of the CIRCUS system, but MRML is also used by the VIPER system developed at the University of Geneva. The PHP interface built by the Vision and

Client	Server
<xml session-id=173> <configure-session> <algorithm algorithm-id=12 collection-id=1 /> </configure-session> <query-step query-step-id=1 result-size=16 query-type=query query-paradigm-id=2> <CIRCUS-browsing-query browse-by=keyword> horse </CIRCUS-browsing-query> </query-step> </xml>	<xml> <query-result service-id=lcavpc52.epfl.ch> <query-result-element-list destination=default> <query-result-element query-result-element-type=1 calculated-similarity="5/5" URI="http://lcavpc52.epfl.ch/CIRCUS/COREL_CDs/502034.jpg" /> <query-result-element calculated-similarity="5/5" URI="http://lcavpc52.epfl.ch/CIRCUS/COREL_CDs/502071.jpg" /> ... </query-result-element-list> <query-result-element-list destination="Overview Panel"> <query-result-element query-result-element-type="CIRCUS-set1''> 102 102 102 102 102 102 102 102 102 102 102 102 102 102 102 102 </query-result-element> </query-result-element-list> </query-result> </xml>

FIGURE 3.6: The client configures the previously opened session and performs a simple query step (from scenario of Figure 3.2).

Multimedia Lab is illustrated in Figure 3.7. The other client software that runs as batch processes is also available and has been used in the Benchatlon effort Beretta and Marchand-Maillet (2001).

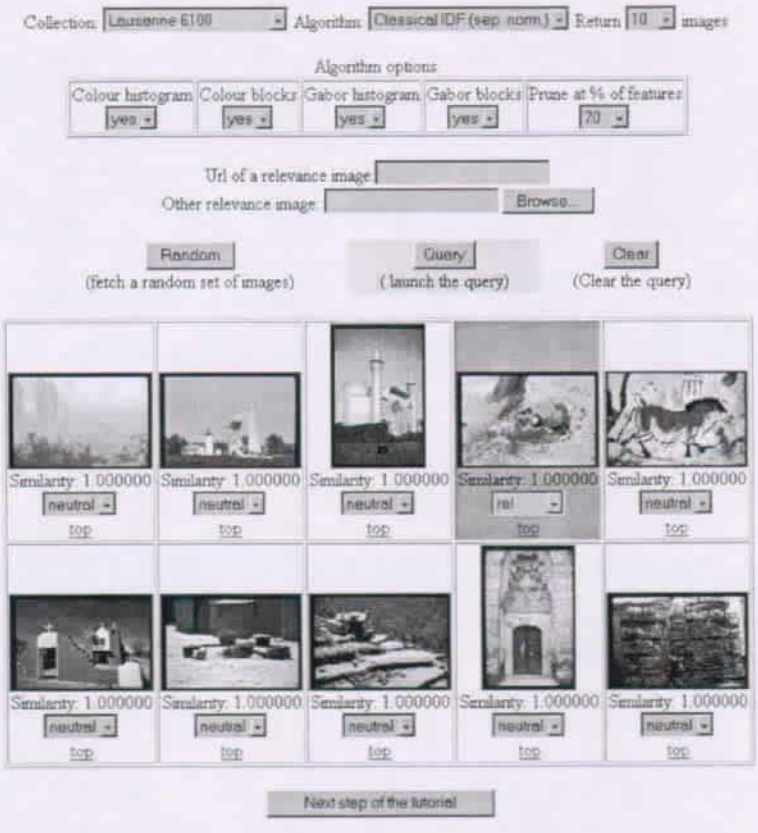


FIGURE 3.7: The PHP WWW interface to the Viper server. Indifferently the server can communicate with the CIRCUS client or the PHP interface.

3.6 Summary and discussion

This chapter introduced and specified the communication protocol that is the binding force between the retrieval method and the user interface. This protocol was also used as a data exchange means among the various components of the retrieval system. The advantages brought by the adoption of this protocol have been discussed: flexibility, extensibility, reusability and intelligibility.

The future research directions include the unification of the protocol extensions into the current specifications. A further important aspect to consider is the extension of the protocol to other media types like video and audio.

3.A MRML details

This appendix describes in some detail the key aspects of MRML. The official web site `http://www.mrml.net` gives additional information and offers the software for download.

The three main topics we present are: the CIRCUS extension to MRML, the MRML query paradigms, and MRML property sheets.

3.A.1 CIRCUS extension to MRML

The extensions to the standard MRML framework necessary for CIRCUS are two-fold:

- Inter-component communication messages used for exchanging data during feature-extraction and management
- Client-server messages that exploit additional CIRCUS functionality.

Inter-component messages

In Table 3.3 we present a review of the extensions to the standard MRML tree used by the interacting components of our architecture. For layout reasons we omit the common prefix `CIRCUS-` of all the message names.

Table 3.3: MRML extensions for cleint-server communication.

Message	Description
<semantics/>	This message can be contained in the data of any of the other CIRCUS extensions. It has optional arguments **language** and **scope**. The first allows for internationalization and the second can take one of two values : "local" or "repeat", for current message or for using a single <semantics/> message for all following CIRCUS extensions of the same type, respectively. It's own data area contains a free text. This message is used either as a comment for the human supervisor, or as an annotation element in a query.
<metadata/>	This message encodes the document file properties in form of attributes, like **creation**, **modification** and **ingestion** dates, **URI**, **size**, **format**, **owner**, **source**, etc.
<feature/>	It is used to convey the extracted features among the document analysis and feature management components. It has **name**, **type**, **producer** and **consumer** as mandatory attributes, and **dim** as an optional attribute. The **producer** attribute identifies the component that produced the feature, and the **consumer** attribute can contain either a destination component name or the generic class of components that should be fed with this message (see discussion below). The understood **type**'s of the feature are "color", "texture", "shape" and "annotation". Other types could be used for other media or other applications. The data within the scope of the tag is a character-based representation of the feature values.
<term/>	This message is output by the first phase of the feature management components, based on the input <document/>, <vocabulary/> and <collection/> messages. It has **name** and **composition** as mandatory attributes while **occurrences** is optional. **composition** identifies the pattern of composition of the term and is a string representation of the binary combination number. The **occurrences** encodes the number of occurrences of the <term/> in a <document/>. The data section is usually empty, but can contain a **semantics** message.
<vocabulary/>	Is a composting message that groups the identified <term/>'s of a <collection/>. It has only **name** and **type** as mandatory arguments. The data section must contain a <vocgen/> message and a series of <term/> messages.

Table 3.3: (continued)

Message	Description
<vocgen/>	This message specifies the type of vocabulary generation that is to be applied to a collection of documents. It is saved in the resulting vocabulary for future reference. The attributes **name**, **collection-name**, **type** are mandatory. The data section depends on the **type** attribute and can contain either a series of <term/>'s (fixed vocabulary setting) or of <feature/>'s in the evolving vocabulary setting.
<document/>	This message is encodes a document. It has **href** and **status** as mandatory attributes which give the URI and the processing status of the document. The data section is dependent on the **status** attribute and can contain a mixture <feature/>, <term/> and/or <category/> messages. It can also contain a <meta-data/> message.
<collection/>	This compositing message groups a set of <document/> messages into a collection. It's mandatory attributes are **name**, **type** and **status**. The last two determine the composition of the contained documents. The values "InProgress", "Unindexed" and "Raw" for this last attribute allow for documents without a term list to be included in the data section, the value **"Indexed"** makes <term/>'s a requirement for each contained <document/>. The data section a <vocabulary/> followed by a series of <document/> elements.
<category/>	This message is a hierarchical structure of the categories present in a collection. It can also appear in a <collection/> message. It's attributes are **name** which is mandatory and **size**, **type**, **coordinate*** and **sample***. These last attributes can be repeated and contain a URI to a series of sample documents for the category and their "coordinates" in an overview structure. The data section is composed of other <category/> elements.

Client-server extensions

In Table 3.4 we present a review of the extensions concerning cleint-server messaging to the standard MRML tree.

Table 3.4: MRML extensions for client-server communication.

Message	Description
<get-overview/>	This message instructs the server to produce an overview of a collection, identified by the attribute **collection-id**, using a specific algorithm, specified by **overview-algorithm-id**. The data section is empty.
<overview/>	This generic message encodes an overview of the collection that was required by the client. Its two mandatory arguments are **collection-id** and **overview-algorithm-id**. The body of the message can contain <property-sheet/>, <document/>, <category/>, <feature/>, <term/> and other pure textual data.
<browsing-query/>	This message can be sent by a client contained within a standard <query-step/> message and requires from the server a selection of representative documents or terms for the given elements of the <browsing-query/>. The mandatory attribute **browse-by** specifies the nature of the passed arguments, which are given inside the data part of the message. The optional argument **result-type** can take on the default value "documents" or the two other possible values: "terms" and "both". The data section can contain free text, or a list of <term/>, <feature/>, or <document/> messages as specified earlier.

Table 3.4: (continued)

Message	Description
<combined-query/>	This message is a wrapper message for combining the query elements in a logical expression. The attribute **active**, if present, indicates that the contained <query-steps/> should be regarded as active, whereas the attribute **negate**, again if present, negates the outcome of the query. The data content is a series of <combined-query/> messages. The messages at the same level are interpreted as logical disjunctions, those nested at deeper levels are considered conjunctions.
<color/>	This message carries the color proportions query. Its attributes are: **space**, assumed to be "rgb" but can be "hsv" and "Luv"; **tristimulus** which specifies the three values; and **amount** which can be expressed as either a percentage or a left unspecified.
<sketch/>	This message is as yet not specified.
<texture/>	This message is as yet not specified.

3.A.2 MRML query Paradigms

The variety of query paradigms that exist in CIRCUS can be accommodated through the MRML native notion of <query-paradigm/>. These messages along with the <query-paradigm-list/> message allows server and client in an MRML setup to describe which collection can be searched by which algorithm and in which mode. MRML supports the two standard modes "qbe" and "browse", CIRCUS adds the modes :

"qbs" corresponding to a query by sketch. The <query-element/> messages of such a query will contain <CIRCUS-sketch/> messages.

"qbp" Encodes a query by document properties. The <query-element/> messages of such a query will contain <CIRCUS-metadata/> parts.

"qbc" Corresponds to a query by color. The <query-element/> messages are in this case composed of <CIRCUS-color/> messages.

"qbt" corresponding to a query by texture. The <query-element/ > is composed of <CIRCUS-texture/> messages.

"combined" this final paradigm allows for combined queries. The contents of the <query-step/> will be a single <CIRCUS-combined-query/> element.

3.A.3 MRML Property Sheets

This ability of MRML makes customizable interfaces for the parameter settings of the various algorithms portable. It is a tree-structured representation of the parameters of the algorithm. The user interface can represent these parameters in any way deemed most appropriate respecting the **visible** attribute.

The use of property-sheets in CIRCUS is restricted to the query-options panel for the moment, but in future elaborations the encoding of the CIRCUS overviews into <property-sheet/> messages is considered.

Chapter 4

Characterizing image content: Image analysis and processing

Effective retrieval depends on the abstraction processing of feature extraction and content description. This chapter is dedicated to the characterization of the perceptual content. The general setting of multimedia retrieval has been restricted in the major portion of our inquiry to image+text documents.

We have concentrated on the methodology of retrieval and on the architectural and structural aspects of the retrieval system. Our goal was not to provide a set of novel characterization methods, or state-of-the-art feature extraction algorithms. So this chapter illustrates our choice among the vast number of existing approaches and gives justification of this choice. However, we have made several contributions to some of the below described methods.

The goal of the methods presented below is to extract from an image and associated textual annotation, an abstract representation of the visual content and encode its associated annotation. The image model we consider is a layered or hierarchical model. The image rectangle is composed of a certain number of regions which are projections of physical objects. The regions are composed themselves of macro-pixels in a regularly tessellated multi-resolution framework. Alternatively, these macro-pixels can be of irregular shape, also corresponding to homogeneous sub-regions of a given region.

The image content characterization is carried out in two phases:

Segmentation The first, top-down phase, decomposes the image according to the layered model. This is performed using un-supervised segmentation of the image into non-overlapping regions. The adopted segmentation method, Normalized Cuts, proposed by Jianbo Shi;"nobreakspace -"Malik (2000), was enhanced in order to make it more efficient. In this respect we propose two alternative speed-up techniques that allow comparable effectiveness in image segmentation results, with a much higher efficiency.

Characterization The second phase creates a synthetic description of the three basic visual aspects of each identified region: color, texture and shape. Additionally, if localized content semantic information is available, we associate it to the region. We discuss the chosen representations and present a brief study of their intrinsic effectiveness.

The way the extracted information is then used for retrieval purposes is the subject of Chapter 5 and Chapter 6.

This chapter is organized in four parts: first Section 4.1 presents a review of related work in image content characterization, especially for the afore mentioned visual aspects. Section 4.2 discusses the problem of global versus local characterization. Then Section 4.3 presents the image segmentation method along with experimental results and discussion. Section 4.4 exposes the four content characterizations: color, texture, shape and semantics. It also presents the performance of the characterization methods for simple discrimination and retrieval. In the closing Section 4.5 we summarize our findings in image content characterization and discuss the drawbacks of our approach. We give some hints for the alleviation of the incurred drawbacks and for future research. In Appendix 4.A additional information on Normalized Cuts is given.

4.1 State of the art in feature extraction and processing

By far this is the most studied aspect of image retrieval. We will describe in the following chapters the precise references that we use for each visual content characterization.

In Table 4.1 we refer just to a few surveys of the rich literature that exists covering image segmentation; color; shape; pattern and texture description; spatial relationship characterization and other aspects related to the extraction of visual content.

Table 4.1: The basic literature for image content characterization.

General surveys	
(Smeulders *et al.*, 2000)	This general and rich review presents all aspect related to image retrieval and is useful as starting point. It reviews more than 200 published papers.
(Yang *et al.*, 2002)	A survey targeted at a specific application: face detection, but it also gives many useful comments on color, shape and texture characterization.
Image Segmentation	
(Lucchese and Mitra, 1999*b*)	This review presents both segmentation and color issues related to shape segmentation.
Color characterization	
(Manjunath *et al.*, 2001)	This surely presents the MPEG-7 accepted color and texture features.
(Aslandogan and Yu, 1999)	This general survey presents most aspect of image content characterization, although lacking a detailed description.
(Lucchese and Mitra, 1999*b*)	see above.
Shape description	
(Loncaric, 1998)	A recent survey of shape analysis, it presents the various approaches (global, boundary, scale-space, morphological, etc.) with more than 220 references.
(Tappert *et al.*, 1990)	This survey is targeted at handwriting description and recognition, but it contains a substantial review of research pertinent to shape characterization.
Texture description	
(Randen and Husoy, 1999)	This survey reviews and compares the performance of several different filter-based texture characterization methods.
(Aslandogan and Yu, 1999)	see above.
(Manjunath *et al.*, 2001)	see above.
Spatial relationships	
(Aslandogan and Yu, 1999)	see above.

4.2 Global versus local characterization

Many previous retrieval system were based on global features of the entire image, which is considered as a single region, and thus any characterization that encodes the region's color or texture perceptual content will work also for the entire image. Shape loses meaning though and the spatial disposition of the relevant objects is inexistent. Global approaches usually fail to retrieve image by similar *content*, although they are usually much more efficient, and can indeed be very effective for discriminating duplicate images, or retrieving different versions of the same image taken under slightly varying imaging conditions (scanning resolution, small variations of view-point, illuminant changes).

The purely local characterization entails a higher burden on the retrieval stage of the system, where an adequate matching function must be specified that decides how relevant an image is since typically results are presented as lists of documents ranked by relevance. An image is relevant to a query that describes content at the local level, if it contains one or more regions of similar local content. The process of deciding which of two candidate images is more relevant when they contain a *single* similar region is quite straightforward. In the case of multiple similar regions the decision usually becomes more involved, and weighting schemes must come into play.

Of course the choice between local and global characterization is going to be conditioned by the application domain of the system. Medical image archiving and retrieval systems would probably opt

for local characterization since usually the system is aimed at finding similar phenomena occurring in various documents. Graphic artists and photographers might also be interested on fast retrieval of different photographs of the same scene, in order to decide on the best shot for publication. Other application domains may require both approaches according to the information need.

The method we propose in Chapter 6 deals with this problem in a way similar to standard text retrieval systems. The regions and their characteristics are "translated" to terms that build up a document. Relevance is decided by term matching and weighting based on implicitly created synonymy relations. This general approach allows for the integration of global descriptions of an image as well as for local content characterization, thus providing the advantages of both strategies indifferently.

4.3 Image segmentation with the Normalized Cut method

The first stage in our processing pipe-line is the segmentation or decomposition of the image into its constituent regions. The definition of a region is quite vague. It is either a part of an object, an entire object or even a set of objects. Automatic segmentation as performed by the human visual system is as yet an open issue, our goal is not to provide a new segmentation method, but rather to apply one existing approach in an efficient way. The adopted method, the various improvements thereof and experimental comparisons are described below.

An effective method for unsupervised segmentation of natural images, or perceptual grouping, is the normalized cut (NC) framework proposed in Jianbo Shi;"nobreakspace -"Malik (2000) and successfully used later in (Malik, 2001). This section outlines the method and presents the rationale behind our choice for using it. It also presents two contributions to the framework that increase the efficiency of the method without compromising its effectiveness.

The set of pixels that build up an image are considered by the NC framework as vertices V of a dense graph. Each pixel is connected to a large number of its neighbors. The edges E of the graph carry a weight which is the similarity of the connected vertices. This similarity can be of any application specific type. The goal of segmentation can be expressed in this setting as a graph partitioning problem. We seek a cut in the graph, based on the co-cycle of a given partition, that is minimized, so that total similarity from one partition component to the other is minimal while the within component total similarity stays as high as possible. This partitioning is then recursively applied to extract more than two regions. In order to avoid problems of trivial partitions or of unbalanced partitions due to outliers, a normalized cut value is used instead of the absolute sum.

If we denote the graph $G = (V, E)$, and take a partition in two regions $A \subset V$ and $B = V \backslash A$. We define the cut between A and its complement B as:

$$cut(A, B) = \sum_{u \in A, v \in B} w_{uv}, \tag{4.1}$$

the total connection between the two regions. w_{uv} denotes the weight of the edge connecting u and v. The weighting is selected to reflect spatial and visual similarity among the two vertices. We also define the association, or total similarity, to a subset of vertices C as:

$$assoc(C) = \sum_{u \in C, v \in V} w_{uv}. \tag{4.2}$$

The normalized cut induced by the partition $V = A \oplus B$ is:

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A)} + \frac{cut(A, B)}{assoc(B)}. \tag{4.3}$$

The solution to the partitioning, and thus a first step towards the segmentation, is thus:

$$\hat{A} = \arg \min_{V = A \oplus B} Ncut(A, B). \tag{4.4}$$

Solving the optimization problem of Equation (4.4) is a NP-complete, however good approximate solutions can be found by re-writing the problem and relaxing some of the constraints.

Lets call $\mathbf{W}$ the adjacency matrix of G and $\mathbf{D}$ the diagonal matrix with $d_{ii} = \sum_j w_{ij}$. Let $\mathbf{x}$ be the partitioning vector with $x_i = 1$ when $v_i \in A$ and $x_i = -1$ otherwise. It can be shown (see Appendix 4.A) that

$$\mathbf{y}_0 = \arg \min_{\mathbf{y}} \frac{\mathbf{y}(\mathbf{D} - \mathbf{W})\mathbf{y}^T}{\mathbf{y}\mathbf{D}\mathbf{y}^\mathbf{T}}, \tag{4.5}$$

where $\mathbf{y} = \mathbf{x}\mathbf{D}^{-1}$ is an equivalent formulation to the minimization of Equation (4.4). The function to minimize in Equation (4.5) is a Rayleigh quotient, and solving it corresponds to finding the smallest eigenvalues of the following generalized eigen-system:

$$\mathbf{y}(\mathbf{D} - \mathbf{W}) = \lambda\mathbf{y}\mathbf{D} \quad \equiv \quad \mathbf{y}\mathbf{D}^{-1/2}(\mathbf{D} - \mathbf{W})\mathbf{D}^{-1/2} = \lambda\mathbf{y}. \tag{4.6}$$

The trivial solution $\mathbf{x}_0 = \mathbf{1}^T$, i.e. not partitioning at all, corresponds to the eigenvalue $\lambda = 0$ the next smallest eigenvalue provides the desired solution.

The additional constraint, that cannot be addressed directly, is that $\mathbf{x}$ must have integer elements equal to $+/-1$. By relaxing this constraint, the problem becomes tractable. The necessary adjustment is the following: Once the eigenvector $\mathbf{y}$ is computed, for each of its elements a decision must be made whether it corresponds to a positive or negative argument. Setting a simple threshold of 0 does not produce the optimal solution. Therefore, a gradient descent search of the resulting $Ncut$ is set up with the initial vector partition threshold set to zero. The outcome is the partition minizing $Ncut$.

To solve the segmentation problem, this bi-partitioning process is than recursively applied to each of the identified vertex sets, A and B. The recursion stops when the normalized cut exceeds a maximum $Ncut$ value.

The only "detail" left is the choice of the similarity $\mathbf{W}$. It is usually composed of at least two factors: a spatial closeness and a visual closeness:

$$w_{ij} = s(i,j) \cdot v(i,j). \tag{4.7}$$

The spatial factor s is any bell-shaped function of the distance between nodes i and j. A suitable such function is:

$$s(i,j) = \begin{cases} e^{-\frac{\|\mathbf{p}_i - \mathbf{p}_j\|^2}{\sigma_s}} & \text{if } \|\mathbf{p}_i - \mathbf{p}_j\| \le r \\ 0 & \text{otherwise} \end{cases}, \tag{4.8}$$

where $\mathbf{p}_i$ is the location of node i, r is a vicinity threshold that reduces the density of the graph, and σ_s is a scaling factor.

The visual factor v is a function that should reflect the visual similarity of two nodes. The choice of v is one of the most delicate issues at hand. If we have simple *dis*similarity functions d_i, such as intensity, color or texture distance, we can build a similarity from these using the following formula:

$$v(i,j) = e^{-\frac{g(d_1,\ldots,d_n)}{\sigma_v}}, \tag{4.9}$$

where g is a combining function such as a weighted sum. The type of difference function used is presented in more detail in Section 4.4.1 for color and Section 4.4.2 for texture. The combining function g we used is the weighted geometric mean, mimicking a distance between points in different spaces (color, texture etc.):

$$g(d_1(i,j),\ldots,d_n(i,j)) = \sqrt{\alpha_i d_i(i,j)^2 + \cdots + \alpha_n d_n(i,j)^2}. \tag{4.10}$$

The two scaling factors σ_s and σ_v are chosen to make the visual and spatial components commensurate. This is necessary in order to avoid numerical ill-conditioning of the matrix $\mathbf{W}$, with extremely different non-zero values.

The normalized cut solution to segmentation, though elegant, is a rather computationally expensive method. We introduce two alternative formulations that significantly enhance performance, while maintaining the same type of effectiveness.

4.3.1 Multi-resolution normalized cut

Multi-resolution normalized cut (MR-NC) processes the image in a multi-resolution framework. Each pixel is averaged with its nine immediate neighbors into a macro-pixel. This is recursively repeated, forming a factor 3 pyramid. This choice will become clear later on.

The standard NC is run at a level of resolution that can be treated in reasonable time, on desktop computers this means the number of macro-pixels is of order $O(10^3) - O(10^4)$. In other words we are limited to images of the size 100 x 100 macro-pixels. This first phase produces a coarse segmentation.

For each boundary among two regions the macro-pixels are refined to the next higher resolution. A new NC graph is established only for the macro-pixels of this border area. One step of the NC

algorithm bi-partitions the boundary area macro-pixels and refines the boundary. The eigen-system solver can be initialized with an initial guess, in this case given by the coarser segmentation. The quality of the guess can significantly improve the computation times. This process is schematically presented on Figure 4.1.

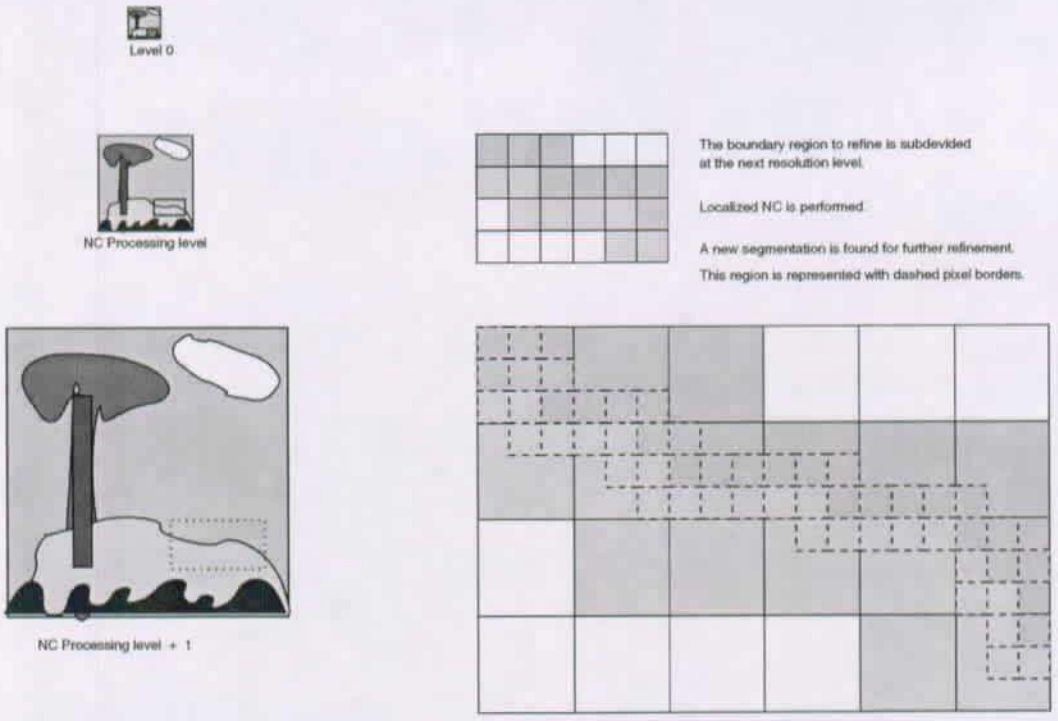


FIGURE 4.1: The original image is processed in a factor three pyramid. Normalized cut is performed on a tractable level L_0. The boundary regions are recursively refined by a localized NC to produce a finer boundary representation.

For junctions of higher order, i.e. separating $n > 2$ regions, several steps of NC are used to obtain at least n regions. Each configuration is examined, and the optimal one chosen. The optimality is based on the resulting $Ncut$ value. Unfortunately, this procedure does not, in and of itself, guarantee a contiguous solution. However, in all our experiments this was the case. Intuitively in the MR pyramid, build by approximation, a boundary at a given level is also present at roughly the same location at a higher resolution level.

This boundary refinement process is then iteratively repeated on the higher resolution layers, until the original pixels are considered themselves. The interiors of the originally segmented regions are simply carried over from one step to the next.

After the initial segmentation, no region will be split and no regions will be merged, i.e. we are stuck with much the same segmentation, simply refining the contours. From the feature extraction point of view, this implies that any statistical changes will be minor, since the interior pixels of a region are by far more numerous than the border pixels.

4.3.2 Watershed normalized cut

Yet another way to enhance performance is the method we call watershed normalized cut (WS-NC). This enhancement starts out by pre-processing the image into a set of irregularly shaped patches. The pre-processing is carried out in three steps:

1. A Laplacian of Gaussian (LoG) filtering is performed to extract local edge energy.
2. The LoG filtered image is then normalized and inverted.
3. The watershed transform (Vincent and Soille, 1991) is applied to produce a fine tessellation (weak segmentation) of the image.

A watershed[1] is a small patch in the image such that any path traversing from one watershed to its neighbor has a maximum on the boundary. This pre-processing, and its results are illustrated on Figure 4.2, for an enlarged portion of one of our test images.

[1]The name comes from a topographical analogy: a drop of rain falling on the "left" side of the boundary will end up in the "left" watershed and vice versa.

FIGURE 4.2: Enlarged image region and pre-processing for watershed normalized cut. Top left: the original image region enlarged for clarity. Top right: after applying Laplacian of Gaussian filtering and inversion (the levels have been normalized to maximum range for visualization). Bottom left: watershed transform. Bottom right: The neighborhood structure of the NC graph for the central watershed.

The watersheds are then considered as the vertices of the normalized cut graph, their location being given by the centroids. Figure 4.2 also shows the graph structure as seen from the node roughly at the center of the cut-out. The rest of the Normalized Cut algorithm is carried out as above, substituting similarity functions based on pixels with ones based on small regions (the watersheds).

Once a labeling is achieved, a Delaunay triangulation, and its dual Voronoi tessellation, are used for re-forming a complete image segmentation. The watershed boundary pixels are assigned to the more similar adjacent region. As we wish the segmentation to preserve the natural separations, like edges, the watershed boundaries should coincide with image edges. This explains our choice of LoG filtering prior to the watershed transform.

The number of watersheds is dependent on the data, and on the choice of the LoG filter "width". Choosing a wide filter yields only the strongest edges thus resulting in large watersheds. Figure 4.3 shows a variety of widths on the same example presented in Figure 4.2. Since the complexity of the algorithm depends on the number of nodes in the graph, it cannot be directly computed. It is, however, upper bounded by half the complexity of raw normalized cut. In fact, in the worst case, i.e. using 4-connectivity and a checkered image, the number of watersheds cannot exceed half the number of pixels. Experimental evidence shows this approach to be at least an order of magnitude faster than raw NC.

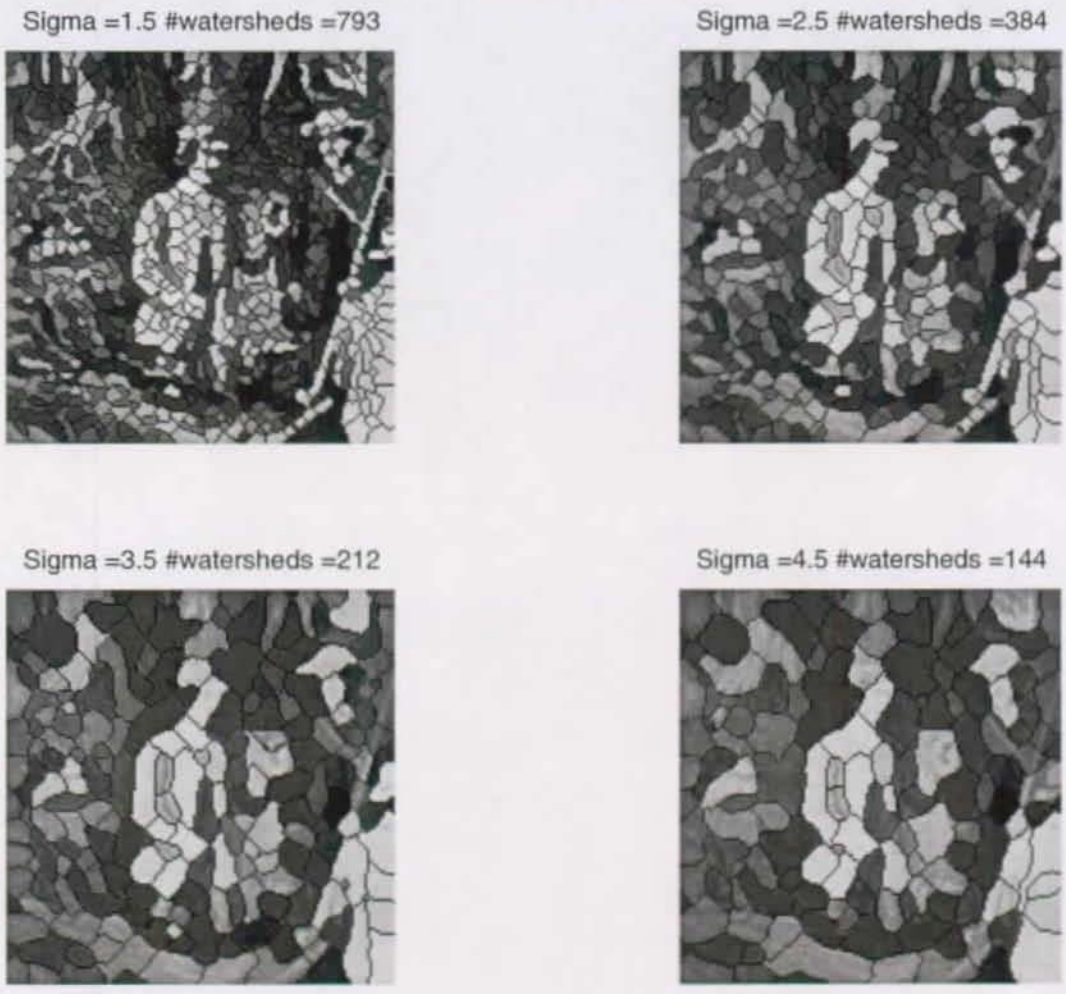
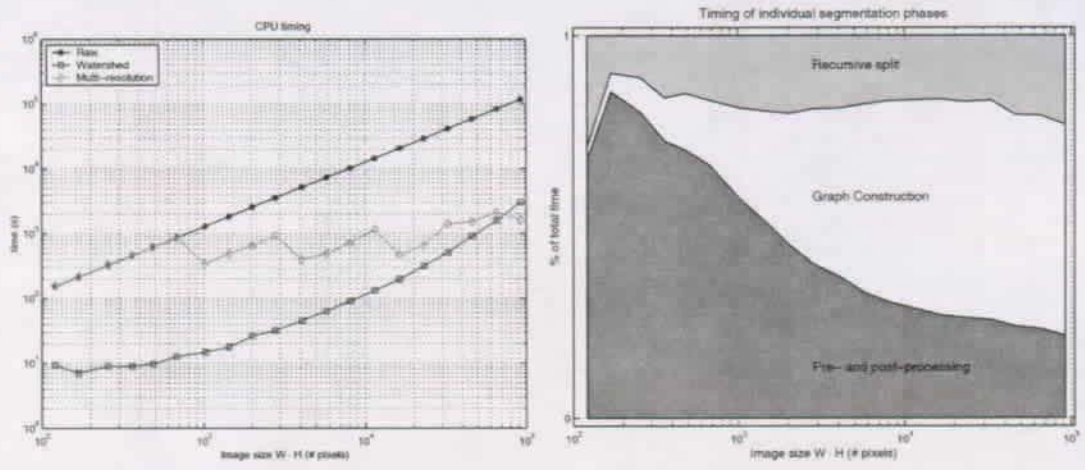


FIGURE 4.3: Different widths for the LoG filter prior to watershed transform.

Performance evaluation A simple optimization of the WS-NC was *not* used in the following tests, namely the adaptation to the image resolution of the widths of the LoG filter kernels. In practice, the number of watersheds being directly proportional to the width of these kernels, according to the number of pixels in the image, we can achieve an almost constant number of watersheds as resolution increases.

Figure 4.4a) shows plots of the execution time for the same image scaled to different sizes. The three NC variants are compared. We see that the behavior of MR-NC is dictated by the choice of processing level. WS-NC maintains a similar behavior as the raw NC variant, however, it is at least an order of magnitude faster. On Figure 4.4b) we plot the split timing for three phases of the computations. i) Pre and post-processing phase (pyramid creation, watershed pre-processing, re-labeling); ii) NC Graph weight computation and management; iii) Recursive graph-splitting. In Table 4.2 we analyzes the complexity of the three approaches in terms of the three above mentioned phases.

Measuring effectiveness is always a difficult task in the segmentation application. We present in Figure 4.5 and Figure 4.6 some experimental segmentation results. In all cases the weighting we used is the combined spatial-visual weighting based on color and texture. The dissimilarity functions used are presented in the following sections.



(a) Absolute timing for the three methods

(b) Split analysis of the execution complexity

FIGURE 4.4: The comparison of execution times for the three normalized cut variants. The proportion of total CPU time for the three processing phases, averaged over all three algorithms and over 8 different images. Notice the saw shape of the MR-NC curve, this is due to the selection of the processing level.

Image	NC			MR-NC			WS-NC		
	G	S	T	G	S	T	G	S	T
1 (256 × 384)	—[a]	—	—	1981	91	2228	1755	101	2091
2 (256 × 384)	—	—	—	1813	87	2056	1801	91	2126
3 (384 × 256)	—	—	—	1935	94	2185	1808	100	2143
4 (256 × 384)	—	—	—	1969	95	2220	1914	132	2280
5 (90 × 90)	11230	234	11472	893	76	1075	710	74	844
6 (90 × 90)	12264	239	12510	936	83	1125	742	74	913
7 (90 × 90)	11138	237	11382	931	78	1111	703	77	879
8 (90 × 90)	11380	249	11181	902	80	1020	702	74	840

[a]The size of the image was too large for testing the raw normalized cut method.

TABLE 4.2: The execution times (in seconds) comparing among NC, MR-NC and WS-NC. **G** is the graph construction, **S** the recursive split and **T** the total execution time respectively. The implementation is in Matlab, running on an Intel Pentium II processor at 450Mhz.

FIGURE 4.5: Sample segmentation results comparing the three variants.

FIGURE 4.6: Sample segmentation results on six test images. The red borders are the region contours.

From these results we can conclude that:

- Raw normalized cut is the slowest but most accurate method. Notice the accuracy difference on Figure 4.5 with the other two methods.
- Multi-resolution and watershed variants are of approximately the same complexity, with WS in general slightly more efficient.
- Choosing adequate watersheds (edge-based) we see that WS-NC produces slightly more natural segmentations than MR-NC.

These experiments have led us to adopt the watershed normalized cut as segmentation method.

4.4 Region characterization

Once an image has been segmented into its constituent regions[2], we need an effective means for describing the visual characteristics of the regions. We describe here the three basic aspects we wish to capture: color, texture and shape. Additionally, we give some hints on exploiting semantic information associated with each region.

4.4.1 Color Characterization

In order to represent colors in digital form, one has to decide on the color space to use. According to the application or the properties that are to be extracted, each color space offers different advantages. The storage formats most commonly use a variation of the RGB color space. This is due to historic reasons, mainly linked to display technology. Among the standard representations it is probably the least indicated for color processing at the perceptual level. Other more perceptually uniform[3] spaces like the CIE Luv, or opponent rgb color space are much more suited to the purpose.

Transforming a tristimulus value from one space to another is a well documented subject in the literature (see (Pratt, 1991, Chap. 3) for a coverage). Unless explicitly stated otherwise, we have used the CIE Lu^*v^* color space for all experiments. The reasons for this choice are: the de-correlation of luminance L and chromaticity u^* and v^* components and good response in terms of perceptual uniformity. The transformation from RGB is done in three steps:

[2]Based on the choice of similarity to use in the Ncut framework, the segmentation results can be very different.

[3]Perceptual uniformity means that there exists a simple formula to compute a color similarity/dissimilarity, which is proportional to a standard observer response.

1. The raw RGB values $(R, G, B) \in [0; 1]^3$ are transformed first to the sRGB space:

$$(R_s, G_s, B_s) = (\gamma(R), \gamma(G), \gamma(B)) \tag{4.11}$$

$$\gamma(x) = \begin{cases} x/12.92 & \text{if } x \leq 0.04045 \\ \left(\frac{x+0.055}{1.055}\right)^{2.4} & \text{otherwise} \end{cases}$$

2. Then the CIE standard observer tristimulus values $(X, Y, Z) \in [0; 100]^3$ are computed:

$$(X, Y, Z) = \cdot \begin{bmatrix} 41.24 & 35.76 & 18.05 \\ 21.26 & 71.52 & 7.22 \\ 1.93 & 11.92 & 95.05 \end{bmatrix} \cdot \begin{bmatrix} R_s \\ G_s \\ B_s \end{bmatrix}. \tag{4.12}$$

3. Finally the Lu^*v^* values are computed. This computation requires the side information of the illuminant chromaticity. We distinguish two cases, natural and artificial lighting. When this information was available from the image annotation it was used (e.g. "outdoor", "landscape photo" — "indoor", "object", "painting"). For the natural lighting we chose the CIE A and for artificial lighting CIE D65 respectively. The formula for $L \in [0; 100], u^* \in [-200 : 200]$ and $v^* \in [-500; 500]$ is:

$$\begin{aligned} L &= \lambda(\tfrac{Y}{Y_l}) \\ u' &= \tfrac{4X}{X+15Y+3Z} \\ v' &= \tfrac{9Y}{X+15Y+3Z} \end{aligned} \tag{4.13}$$

$$\begin{aligned} u^* &= 13L(u' - u'_l) \\ v^* &= 13L(v' - v'_l), \end{aligned} \tag{4.14}$$

$$\lambda(y) = \begin{cases} 903.3y & y \leq 0.008856 \\ 116\sqrt[3]{y} - 16 & \text{otherwise} \end{cases}, \tag{4.15}$$

where Y_l is the intensity of the illuminant and u'_l and v'_l the values obtained by substituting the illuminant (A or D65) tristimulus values into Equation (4.13).

Having chosen a "good" color space for representation, our next goal is to describe a set of pixels P. If the size of the set is large, or the variation of the color values significant, an adequate representation, widely used and studied, is the histogram:

$$H(c) = |\{p \in P : p = c\}|, \tag{4.16}$$

where c is a given color. If we normalize $H(c)$ by the size of P, we can consider it to be a probability density function (pdf) of the discrete random variable that expresses the color of a given pixel in P:

$$p(c) = \frac{H(c)}{|P|}. \tag{4.17}$$

Of course H cannot be evaluated for all possible colors, so a discrete color set must be chosen. Alternatively one can choose to ignore the luminance information and consider only the chromaticity histogram of P. Figure 4.7 shows a histogram of a sample image region for: i) a fixed palette of colors and ii) a quantized chromaticity plane.

It has been shown in (Puzicha *et al.*, 1999) that a more robust representation than the histogram

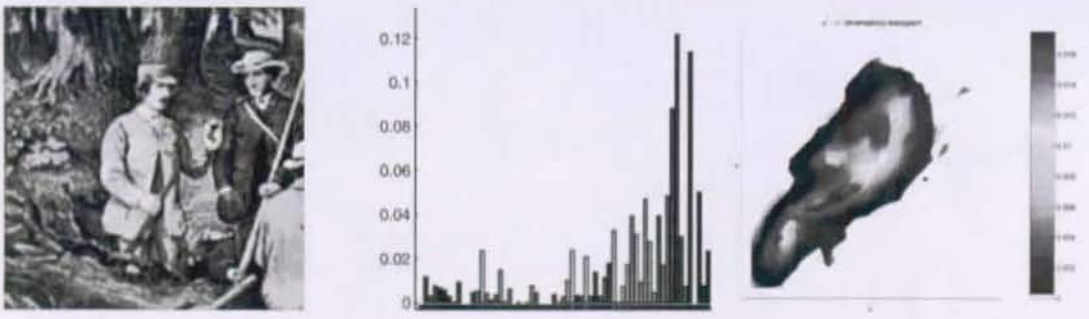


FIGURE 4.7: An image region and two color histogram characterizations.

(in 1-D cases obviously) is the cumulative histogram:

$$C(c) = \sum_{k<c} H(c). \tag{4.18}$$

Again, considering a normalized version of C we arrive at the cumulative distribution function (cdf) of the discrete random variable:

$$\mathrm{CDF}(c) = \sum_{k<c} p(c). \tag{4.19}$$

Building further on top of this probabilistic interpretation, we can use a moment expansion of the pdf:

$$m(l) = \sum_{c \in C} p(c) c^l. \tag{4.20}$$

$m(p)$ is the l-th order moment. The infinite set of moments, $l \in [0; \infty]$, uniquely defines the pdf (see (Jain, 1989, p. 378) for a sketched proof). $m(0) \equiv 1$ for any pdf. The central moments

$$\mu(l) = \sum_{c \in C} p(c) \big(c - m(1)\big)^l, \tag{4.21}$$

have concrete statistical interpretations; the first 4 are the mean, variance, skewness and kurtosis respectively. The moment characterization is an even more robust description than the cdf.

The extensions to chromaticity histograms, based on the two chromatic components of the Lu^*v^* space u^* and u^*, is straightforward. The moment expansion of Equation (4.21) becomes:

$$\mu(k,l) = \sum_{c \in C} p(c) \big(c_1 - m(1,0)\big)^k \big(c_2 - m(0,1)\big)^l. \tag{4.22}$$

This moment representation is also beneficial since it is compact and robust.

If P is a small region, or of relatively homogeneous color, the histogram or higher order moments are no longer an economic characterization. A first, statistically sound, solution is to store an approximation of the region color, like the average or median color. The higher order statistics can be used, however, in expected value they are almost certainly equal to zero, since the region is homogeneous to start with.

To recapitulate, for characterizing the color of an image component, macro-pixel, region or entire image, we differentiate two cases:

Large or multi-colored component If the size of the component is above a certain threshold τ_S or the color variance larger than σ_c^2 we can use either the raw histograms H and C or a few of their moments.

Small or uniform component Otherwise, under the homogeneity assumption we store only a representative color: either the average, or in a more robust fashion, the median color.

In order to establish a difference estimation between two sets of pixels P_i and P_j we use the following rule:

- If both sets are described by histograms:

$$d(P_i, P_j) = \frac{1}{|C|} \sum_{c \in C} |H_i(c) - H_j(c)|. \tag{4.23}$$

- Otherwise transform both descriptions into moment form $\mathbf{m}_{\{i,j\}} = (\nu(1), \ldots, \nu(n))^T$ and use:

$$d(P_i, P_j) = ||\mathbf{m}_i - \mathbf{m}_j||. \tag{4.24}$$

4.4.2 Texture Characterization

The variance of the color values inside an image region is not truly random, rather it follows some hidden formation rules. The patterns that appear are often pseudo-periodic, spatially correlated or otherwise compositionally coherent. The goal of texture analysis or characterization is to model these hidden rules or at least to encode synthetic information about the structured nature of the patterns.

Again we have investigated this area just superficially, in order to provide a robust and usable texture characterization scheme. We directly present the model we use and some results. For more adequate texture characterization refer to (Do and Vetterli, 2002).

Most texture analysis models are centered around an image transform that highlights features of the signal not only in the time/space domain, but also in the frequency/scale domain. Classical Fourier analysis and short-time Fourier analysis have given way to a more practical, more versatile and more effective set wavelet based transforms.

FIGURE 4.8: Original image region, its wavelet transform and the significant coefficients positions.

The wavelet transform produces a coarse representation of the signal and a series of approximations. Figure 4.8 shows the transform of an image region with enhanced range for printing. The different sub-bands are associated with the vertical, horizontal and diagonal responses in the image. Our goal is to capture the information in the sub-bands, or at least to describe the distribution of their wavelet coefficients. These scale-space linear transforms, like the wavelet transform, have some of the ideal features for texture analysis, among their many other applications. They are fast to compute, they offer insight into both the spatial relations among the patterns (where the patterns recurs), as well as to scale relations (at which size a pattern recurs). They also leave the designer a great deal of control over which properties of the signal to extract.

The simplest description is a moment based description of the distribution of the coefficients. We first approximate the distribution by quantizing the coefficients and constructing a discrete empirical pdf. By construction, the coefficients have zero mean, so higher order moments are the only interesting ones. The first descriptor computed is for each sub-band s of the region:

$$\mathbf{v}_s = (\text{var}(s), \text{skew}(s), \text{kurt}(s))^T. \tag{4.25}$$

$\mathbf{v}_s$ offers no indication of the spatial location of significant coefficients. We consider a threshold value τ and the set of coefficients greater than τ:

$$\mathcal{H}_s = \{c[k,l] \in s : |c[k,l]| > \tau$$

. We want to characterize the random variable $X = (k,l)$ that gives the location of the significant coefficients, so we compute a moment description of X as in the 2-D color histogram case:

$$\mathbf{h}_s = \left(m(0,0), m(0,1), m(1,0), \mu(1,1), \mu(1,2), \mu(2,1), \mu(2,2)\right)^T, \tag{4.26}$$

where μ are normalized central moments.

For establishing a difference estimation between two regions we use the concatenation for all sub-bands $\mathbf{t} = (\mathbf{v}_1; \mathbf{h}_1; \cdots \mathbf{v}_s; \mathbf{h}_s)$. Then we use the Mahalanobis distance:

$$d(\mathbf{t}_i, \mathbf{t}_j) = \left[(\mathbf{t}_i - \mathbf{t}_j)^T \cdot \mathbf{C}^{-1} \cdot (\mathbf{t}_i - \mathbf{t}_j)\right]^{1/2}, \tag{4.27}$$

where $\mathbf{C}$ is the covariance matrix of $\mathbf{t}$. This choice of metric alleviates somewhat the highly correlated and incommensurate nature of the different descriptor components.

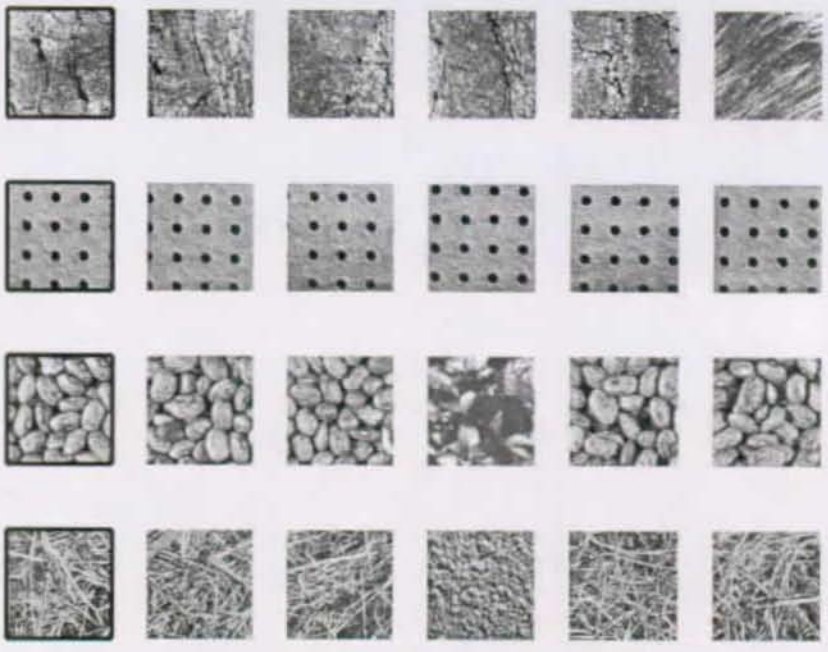


FIGURE 4.9: Results for a simple texture discrimination model: The 3 most similar images to the first in each row are shown . The set contains 640 texture samples.

On Figure 4.9 we show a sample of the effectiveness of our texture description for a set of square images from the (Vistex, 1995) collection. On Figure 4.10 we have also plotted the precision and recall curve (see Appendix 6.A for definitions).

FIGURE 4.10: The precision-recall diagram plots the averages across all the different texture samples of the simple texture model.

4.4.3 Shape Characterization

The final visual intrinsic characteristic of a region is its shape. If the segmentation produces a region that closely fits the object projection, this shape also characterizes the object itself.

There are many ways to describe a shape, we present here the approach we have adopted. The main criteria used for the selection among the many available methods (see Section 4.1 for references) was that we wanted a simple method to implement. Our purpose still being more a proof of concept rather than a top-notch feature extraction and description method.

The basic setup is depicted on Figure 4.11

Each region's external boundary[4] is extracted and labeled. We form the sequence $\rho[i]$ with the distance from the region centroid to boundary pixel i. The description of this contour is performed using moment invariants. First we define the r-th order contour sequence moment m_r and central moment μ_r:

$$m_r = \frac{1}{N}\sum_{i=1}^{N}\left[\rho[i]\right]^r, \tag{4.28}$$

[4]Normalized cut method can produce regions that are not contiguous and/or that contain holes. In this latter case the perimeter is considered and the holes are ignored.

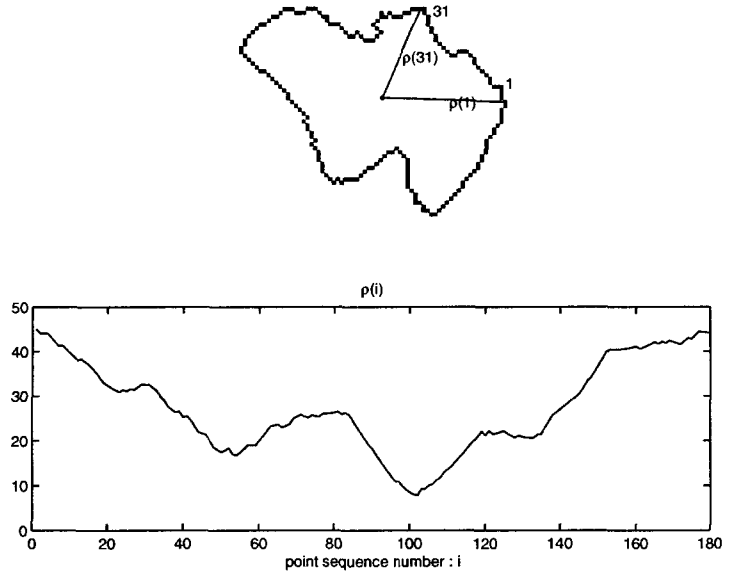


FIGURE 4.11: Shape characterization setup. A region contour and the sequencing along the border of the distance from centroid. Below the flattened-out distance profile.

$$\mu_r = \frac{1}{N}\sum_{i=1}^{N}\left[\rho[i] - m_1\right]^r, \tag{4.29}$$

Gupta and Srinath (1987), present a set of translation-, rotation and scale-invariant normalized contour moments $\bar{m}_r$ and $\bar{\mu}_r$:

$$\bar{m}_r = \frac{m_r}{(\mu_2)^{r/2}}, \tag{4.30}$$

and

$$\bar{\mu}_r = \frac{\mu_r}{(\mu_2)^{r/2}}. \tag{4.31}$$

Gupta and Srinath, claim that probability of error is reduced using these contour moments, with respect to using area based moments.

We decide to describe a region i with a vector:

$$\mathbf{s}_i = (\frac{\sqrt{\mu_2}}{m_1}, \bar{\mu}_3, \bar{\mu}_4, \ldots \bar{\mu}_9)^T. \tag{4.32}$$

Region shape differences are measured by the dissimilarity function:

$$d(P_1, P_2) = ||\mathbf{s}_1 - \mathbf{s}_2||. \tag{4.33}$$

For evaluation, two test sets are considered: i) a collection of 512 images containing one of the 9 geometric shapes scaled, rotated and pasted at random in the image, ii) a collection of 400 images containing one natural object among 20 pasted on a noisy background after scaling and rotation. Figures 4.12 to 4.14 show a sample of performance and precision-recall curves, for the geometric- and natural-shape sets.

Even though the effectiveness is not perfect, the moment description discriminates among shapes rather well. In Chapter 6 we explain in what circumstances this shape information will be used.

4.4.4 Text Characterization

Textual or semantic annotation is the final characteristic that we would like to associate with image regions. If the user is willing to provide a semantic label to the identified image region, then our problem is solved. If however the images are processed in large numbers, and manual assistance from the user is not available, then some automatic means must be used. The Latent Semantic Indexing retrieval method discussed in Chapter 5 and Chapter 6 allows us to solve this problem, even if not completely. Without going into details, the method can establish synonymy relations among visual and semantic aspects of the images in the collection. So when retrieving regions of an image that were not previously annotated, the method can also retrieve the most "synonymous" semantic labels, and

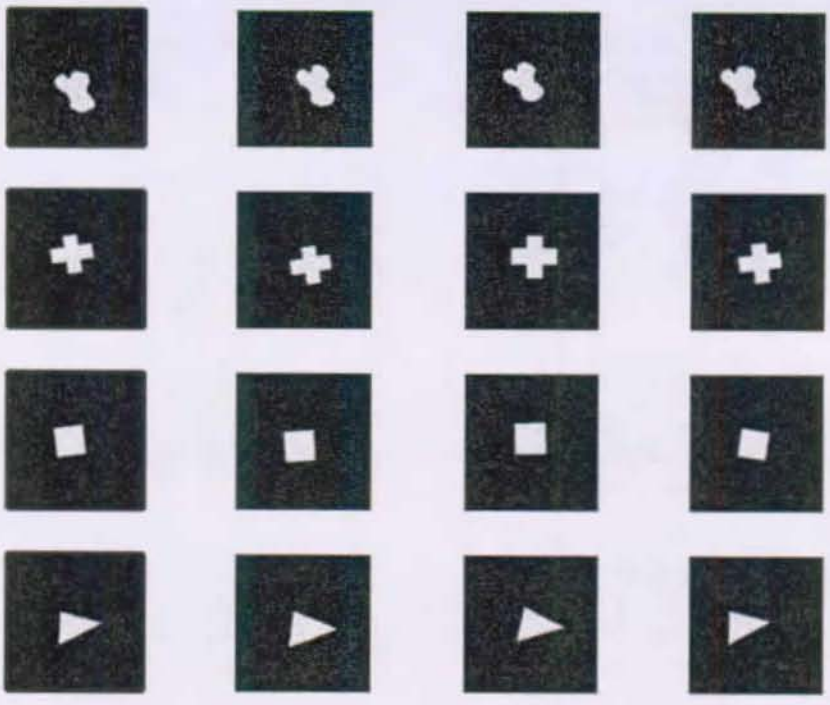


FIGURE 4.12: Shape matching on geometric shape collection. A few most similar results the first shape in each row.

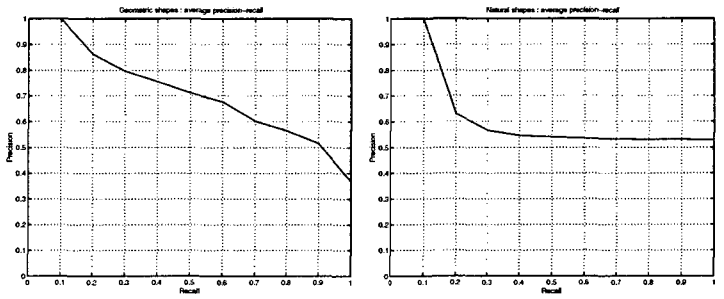
FIGURE 4.13: Shape matching on natural shapes collection. A few most similar results to the first shape in each row. The collection is 400 images of 20 shapes on textured background.

similarly when issuing queries for semantic labels, regions never annotated can be retrieved.

For the moment, suffice it to say that whenever possible this semantic information should be stored in tight association to the image region.

4.4.5 Global characteristics

When an image is being analyzed for indexing, the color, texture and annotation characteristics of the entire image, considered as a single region, are also extracted. The way this information is then integrated with local characteristics in the same index is presented in detail in Chapter 6.



(a) Geometric dataset (b) Natural dataset

FIGURE 4.14: The precision-recall curves for shape matching on simple geometric (a) and natural (b) shapes collections.

4.5 Summary and discussion

This chapter presented a series of discussions on image content characterization. Our analysis model was chosen to be hierarchical: the image under scrutiny was first segmented and then its regions where analyzed to capture the color, texture and shape properties.

The efficiency of the chosen segmentation framework, Normalized Cuts, was improved by two approaches that abstract the notion of pixel level segmentation to macro-pixel level segmentation. This approach leads to a reduction of running time by at least an order of magnitude, without sacrificing effectiveness. The local behavior at the region boundaries is sub-optimal, but in the retrieval application domain very accurate boundaries are not a requirement. We rather capture the salient objects approximately, and agree to sacrifice a bit of localization for the benefit of a much faster algorithm.

The subsequent synthetic description of color, shape and texture, was solved by simple and fast algorithms. The performance evaluations and comparisons to standard approaches were presented and shown to be reasonable. We highlight the fact that such basic shape and texture descriptions, as given in this chapter, will usually not be sufficient to discriminate to an acceptable degree the regions of the images in a real-life data collection. Adapting other more specific or more accurate feature extraction methods is one of the major improvements that is still open for future work.

4.A Equivalent formulation to the Normalized Cut optimization

The contents of this appendix are a based on Shi and Malik (2000). The article also presents a proof of the NP-completeness of the problem, as well as discussion of the aspects related to image segmentation.

The Normalized cut optimization problem :

$$\min_{A=V-B} Ncut(A,B) \tag{4.34}$$

can be expressed in terms of the symmetric adjacency matrix $\mathbf{W}$, the total flux per node matrix $\mathbf{D}$ and a vector $\mathbf{x}$ that satisfies:

$$x_i = \begin{cases} 1 & \text{if node } i \in A \\ -1 & \text{if node } i \in B \end{cases}. \tag{4.35}$$

We can write:

$$Ncut(A,B) = \frac{\sum_{(x_i>0,x_j<0)} -w_{ij}\mathbf{x}_i\mathbf{x}_j}{\sum_{(x_i>0)} d_{ii}} + \frac{\sum_{(x_i<0,x_j>0)} -w_{ij}\mathbf{x}_i\mathbf{x}_j}{\sum_{(x_i<0)} d_{ii}}. \tag{4.36}$$

Define k as :

$$k = \frac{\sum_{(x_i>0)} d_{ii}}{\sum_i d_{ii}}, \tag{4.37}$$

and denote by $\mathbf{1}$ the $N \times 1$ vector of all ones.

Since $\frac{\mathbf{1}+\mathbf{x}}{\mathbf{2}}$ and $\frac{\mathbf{1}-\mathbf{x}}{\mathbf{2}}$ are the indicator vectors for $\mathbf{x}$, we have :

$$\begin{aligned} 4Ncut(A,B) = & \frac{(\mathbf{1}+\mathbf{x})^T(\mathbf{D}-\mathbf{W})(\mathbf{1}+\mathbf{x})}{k\mathbf{1}^T\mathbf{D}\mathbf{1}} \\ & + \frac{(\mathbf{1}-\mathbf{x})^T(\mathbf{D}-\mathbf{W})(\mathbf{1}-\mathbf{x})}{(1-k)\mathbf{1}^T\mathbf{D}\mathbf{1}} \\ = & \frac{\mathbf{x}^T(\mathbf{D}-\mathbf{W})\mathbf{x} + \mathbf{1}^T(\mathbf{D}-\mathbf{W})\mathbf{1}}{k(k-1)\mathbf{1}^T\mathbf{D}\mathbf{1}} \\ & + \frac{2(1-2k)\mathbf{1}^T(\mathbf{D}-\mathbf{W})\mathbf{x}}{k(k-1)\mathbf{1}^T\mathbf{D}\mathbf{1}}. \end{aligned} \tag{4.38}$$

Let

$$\begin{aligned} \alpha(\mathbf{x}) &= \mathbf{x}^\mathbf{T}(\mathbf{D}-\mathbf{W})\mathbf{x}, \\ \beta(\mathbf{x}) &= \mathbf{1}^\mathbf{T}(\mathbf{D}-\mathbf{W})\mathbf{x}, \\ \gamma &= \mathbf{1}^\mathbf{T}(\mathbf{D}-\mathbf{W})\mathbf{1} \end{aligned}$$

and

$$M = \mathbf{1}^\mathbf{T}\mathbf{D}\mathbf{1}.$$

Then we can rewrite 4.38 as :

$$\begin{aligned} 4Ncut(A,B) = & \frac{(\alpha(\mathbf{x})+\gamma) + 2(1-2k)\beta(\mathbf{x})}{k(1-k)M} \\ = & \frac{\frac{1-2k+2k^2}{(1-k)^2}(\alpha(\mathbf{x})+\gamma) + \frac{2(1-2k)}{(1-k)^2}\beta(\mathbf{x})}{\frac{k}{1-k}M} + \frac{2\alpha(\mathbf{x})}{M}, \end{aligned} \tag{4.39}$$

since $\gamma = 0$.

Now letting

$$b = \frac{k}{1-k} = \frac{\sum_{x_i>0} d_{ii}}{\sum_{x_i<0} d_{ii}},$$

we have

$$= \frac{(1+b^2)(\alpha(\mathbf{x})+\gamma)}{bM} + \frac{2(1-b^2)\beta(\mathbf{x})}{bM} + \frac{2b\alpha(\mathbf{x})}{bM} \underbrace{-\frac{2b\gamma}{bM}}_{=0} \tag{4.40}$$

$$= \frac{(\mathbf{1}+\mathbf{x})^{\mathbf{T}}(\mathbf{D}-\mathbf{W})(\mathbf{1}+\mathbf{x})}{b\mathbf{1}^{\mathbf{T}}\mathbf{D}\mathbf{1}} + \frac{b^2(\mathbf{1}-\mathbf{x})^{\mathbf{T}}(\mathbf{D}-\mathbf{W})(\mathbf{1}-\mathbf{x})}{b\mathbf{1}^{\mathbf{T}}\mathbf{D}\mathbf{1}} - \frac{2b(\mathbf{1}-\mathbf{x})^{\mathbf{T}}(\mathbf{D}-\mathbf{W})(\mathbf{1}-\mathbf{x}))}{b\mathbf{1}^{\mathbf{T}}\mathbf{D}\mathbf{1}}$$

$$= \frac{[(\mathbf{1}+\mathbf{x})-b(\mathbf{1}-\mathbf{x})]^T(\mathbf{D}-\mathbf{W})[(\mathbf{1}+\mathbf{x})-b(\mathbf{1}-\mathbf{x})]}{b\mathbf{1}^{\mathbf{T}}\mathbf{D}\mathbf{1}}.$$

So, if we set $\mathbf{y} = (\mathbf{1}+\mathbf{x}) - b(\mathbf{1}-\mathbf{x})$, we can see that :

$$\mathbf{y}^{\mathbf{T}}\mathbf{D}\mathbf{y} = \sum_{X_i>0} d_{ii} + b^2 \sum_{x_i<0} d_{ii} = b\mathbf{1}^{\mathbf{T}}\mathbf{D}\mathbf{1}. \tag{4.41}$$

So substituting $\mathbf{y}$ and 4.41 into 4.40 and restating the optimization problem 4.34 we have:

$$\min_{A=V-B} Ncut(A,B) = \min_{\mathbf{y}} \frac{\mathbf{y}^{\mathbf{T}}(\mathbf{D}-\mathbf{W})\mathbf{y}}{\mathbf{y}^{\mathbf{T}}\mathbf{D}\mathbf{y}}, \tag{4.42}$$

with the original conditions changed to:

$$y_i \in \{1, -b\} \quad \text{and} \quad \mathbf{y}^{\mathbf{T}}\mathbf{D}\mathbf{1} = 0. \tag{4.43}$$

The problem as stated in 4.42 is a Rayleigh quotient corresponding to the generalized eigen-system:

$$(\mathbf{D}-\mathbf{W})\mathbf{y} = \lambda\mathbf{D}\mathbf{y}. \tag{4.44}$$

Now since $\mathbf{D}-\mathbf{W}$ is semi-definite positive, all its eigenvectors are orthogonal to each other, and since $\mathbf{1}$ is an eigenvector with eigen-value 0, and $\mathbf{D}$ is diagonal, the second constraint is automatically met by all solutions of 4.44.

Relaxing the first constraint, i.e. that $\mathbf{y}$ takes on only one of two discrete values, is what makes the solution tractable.

Chapter 5

Latent Semantic Indexing

Many approaches have been considered for solving the problem of information retrieval. All of them rely on some model of the information contained in a collection of documents. Latent Semantic Indexing (LSI) (Deerwester *et al.*, 1990) is the method we will describe in more detail in this chapter.

First though we will present the general models used in information retrieval in Section 5.1, then in Section 5.2 we will present LSI from a global and structural viewpoint. In Section 5.3 we give a historical account with pointers to LSI applications. After this, in Section 5.4 we develop in depth the mathematical constructs needed to implement LSI. Section 5.5 points out some practical issues that arise when implementing LSI. In order to clarify some of the assertions made in this chapter we present a toy example of LSI on a small collection of image captions. No visual features have been used to produce this example. Please note that Chapter 6 is specifically dedicated to the application of LSI to image retrieval, and the integration of visual and textual document content into the same indexing structure. Finally Section 5.6 contains a summary and some thought is given to further investigations that seem still open.

5.1 Information retrieval models

Any information retrieval application must model the collection of documents and the building blocks that compose the documents. We propose to group information retrieval models in five broad structural categories. Table 5.1 summarizes the principal characteristics and gives a tentative taxonomy of the models.

Vector space models The Vector space models represent the documents as points in a high dimensional space. Each coordinate corresponds to a term. Thus, the information contained in each document is represented in terms of a set of appearing features. The structural emphasis of this model is on the documents, since from a document representation we can deduce the composing terms. It is well adapted to retrieval by document similarity and allows fairly straightforward manipulations of the search space.

Inverted file models Inverted files or indexes represent the information from a term centered point of view. Here for any term, or basic information element, the structure keeps a list of documents that possess it. It is a more efficient representation than vector space models; it allows simple access to documents relevant to a set of terms. In this model it is a slightly more complicated task to determine document similarity.

Probabilistic models This approach models the information in the collection by assigning to documents a-priori probabilities of being relevant, which is then conditioned on the actual query and the user relevance information. The final likelihood that the document is relevant is thus established during run-time.

Knowledge-, rule, and case-based models In the rule-based, knowledge based or case-based models the information is structured according to the application domain and complex rules encode the relevance to queries based on the constructed information model of each document. These models offer a great performance increase but are hardly flexible enough in general cases. Especially since a large part of the information has to be processed manually for the construction of the model. They are usually supplemented by inference engines, constraint satisfaction engines

or case-based reasoning that allow for the creation of new rules or knowledge based on previous models and user feedback.

Latent Semantic Analysis LSA produces an index structure called a Latent Semantic Index based on a statistical analysis of occurrence patterns. It is somewhat a mid-way solution among the four groups presented above. It tries to offer the advantages of all these models. Precisely how is explained in this chapter.

Table 5.1: A short taxonomy of information retrieval models.

Model	TO	DO	S	K	F	Description
Plain VS	-	+	+	-	+	Plain Vector space models like those described in (White and Jain, 1996*a*) offer high flexibility for similarity searches and for feedback incorporation. They lack however a means for encoding some basic knowledge of the collection.
Plain IF	+	-	+/-	-	+/-	Plain Inverted indexes like those described in (Brown *et al.*, 1994) offer high efficiency for term based queries and outperform other models for simple tasks, they too lack knowledge encoding. The feedback can be provided for relatively easily if a hybrid probabilistic model is used (Cox *et al.*, 1996).
Probabilistic	+/-	+/-	+/-	+	+	Probabilistic models (e.g. (Cox *et al.*, 1996)) usually lie on top of a basic vector space model or inverted file model, so according to which one we have more or less flexibility for document or term retrieval and similarity. They can encode application domain knowledge well and offer a high degree of adaptability based on user feedback.
K-,R- and C-based	+/-	+/-	+/-	+	+	Knowledge, rule and case-based systems can offer good term or document retrieval according to the underlying structure, they often offer poor similarity retrieval but compensate with high knowledge encoding and versatile feedback incorporation.
LSI	+	+	+	+	+/-	LSI (Deerwester *et al.*, 1990) is presented in detail in this chapter.

TO: term oriented, DO : Document oriented, S: allows retrieval by similarity, K: encodes knowledge, F: allows user feedback.

5.2 An overview of Latent Semantic Indexing

Latent Semantic Analysis (LSA) and the resulting indexing method (LSI) starts out by modeling a set of documents, called collection or corpus. The model chosen to represent the information is the term-by-document co-occurrence matrix.

$$\mathbf{A} = \begin{array}{c} \text{terms} \\ \left[\begin{array}{ccc} a_{11} & \cdots & a_{1N} \\ & \ddots & \\ a_{M1} & \cdots & a_{MN} \end{array}\right] \end{array} \text{documents} \qquad (5.1)$$

In other words we store for each term a representation of which documents contain it. From a dual point of view we also store for each document the terms that compose it. As mentioned in the introduction to this chapter, both a document and term centered view are possible, simply by considering the matrix column-wise or row-wise.

As in most information retrieval models the raw information stored in the collection representation (the term-by-document co-occurrence matrix) must be weighted to avoid giving too much importance to terms that appear too often, or biasing the retrieval towards longer or short documents. A weighting transforms the raw term-by-document co-occurrence matrix to a more normalized matrix.

In a large collection of documents from many authors, we encounter problems related to term usage. The most typical problem is synonymy, where two different terms are used to represent the same semantic concept[1]. In a slightly more devious fashion, we encounter polysemic terms, i.e. terms that according to the context or domain of the document may represent different semantic concepts regardless of the same syntactic representation[2]. In order to solve these problems LSI proposes to represent the term-by-document co-occurrence matrix with a lower rank approximation.

The rationale behind is that terms that jointly occur in many documents are related and represent a unique concept. By changing the representational basis we would like these pseudo-synonyms to end up as a unique direction. Conversely if a given syntactic term is used in conjunction with two sets of otherwise non co-occurring terms, we would like its representation to be split into two directions or in a direction distinct, yet close to the direction of both co-occurring sets.

> **Example:** A set of documents contains the terms *patient, doctor* and *surgeon* and another set contains *patient, doctor* and *physician* the terms *surgeon* and *physician* become related, even if they never occur together. User queries containing *surgeon* will return, to a lesser extent, documents where it does not appear, but which contain *physician.*

A lower-rank approximation of a matrix does just that, it "merges" highly correlated data into a single basis vector, and at the same time orients the basis vectors in such a way as to span the different concept directions. LSA thus produces a transformation method of the term-by-document co-occurrence matrix which can accommodate both column-wise and row-wise correlation analysis. The best such approximation is given by the Singular Value Decomposition which is described in Section 5.4. An important aspect is the ability to represent in the same space the "projections" of both terms and documents.

The index is given by the transformed data *and* the transformation method. The querying is performed in a few basic steps:

1. A query is analyzed and transformed into the same representation as the collection.
2. The collection is traversed by comparing the query representation to the representations of the documents and/or terms.
3. Those that are closest, with respect to a given measure, are returned as results.

Either terms or documents can be used as queries and *both* documents and terms can be returned as results. This is a noteworthy advantage over the more conventional methods.

5.3 LSI historical background

LSA has been proposed in the early nineties by Deerwester *et al.* (1990). It has been patented and applied mainly to information retrieval from a corpus of textual documents. The discussion above describes the scenarios where the corpus is relatively stable and fixed whereas users are numerous, each with a varying need for information. LSI has also been used for information filtering (Foltz, 1990). In this situation the corpus is a varying set of documents, like news, entertainment broadcasts, whilst the principal user is one and has a slowly varying information interest. This interest is modeled by an LSI vector and the new documents are "projected" into the LSI model and compared to the user interest vector. If they match they are returned to the user, with possible user interest modifications proposed.

Further applications in the textual domain include multi-lingual retrieval. In this application domain a corpus of documents exists in various languages, some documents are considered to be translations of each-other while others are not. LSI models the concepts contained in the paired (translated) documents and then uses the model to retrieve documents which do not necessarily have a translation

[1] Consider terms in text documents only, then we have situations like *car – automobile*

[2] Again in purely textual documents *tree* can represent both a plant and a data-structure.

in the corpus. Pointers to these applications include (Dumais *et al.*, 1997). A similar approach can be used for automated or assisted thesaurus and dictionary constructions.

We cite a series of articles that all present LSI, modeling its results through probabilistic and other means, and give some indication of the foundations for its effectiveness: (Story, 1996), (Witter and Berry, 1998), (Papadimitriou *et al.*, 1998), (Ding, 1999) and (Hofmann, 1999*b*).

We also refer to the LSI web page which presents the various other applications of LSI in text-based context `http://www.cs.utk.edu/~lsi`.

At a more abstract level, LSI has been used for modeling learning, writer assisting and tutoring, essay grading, conference planning, and many other text-based applications. A comprehensive list of applications and publications can be found at `http://lsi.research.telcordia.com`.

A range of applications that concern us most is the one dealing with multiple media documents. Oddly enough LSI has not been extensively studied in this domain. In (La Cascia *et al.*, 1998), LSI is used to index image captions, then the resulting index is combined with a classical image feature space index and used as an orthogonal complement. This approach although useful for indexing captions does not exploit any LSI feature that links the visual and textual aspects of the data.

In (Kurimo, 1999), LSI is used on top of a speech recognition system to index spoken audio documents. Here again LSI is used only in its textual form, without attempting to represent the aural aspects of the documents.

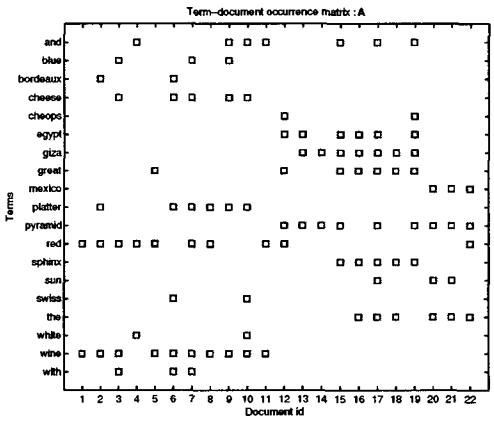
Westerveld *et al.* (2000) presents ideas on image retrieval similar to our own (Chapter 6). There seems never to have been any follow up to that research, and *no implementation or testing.*

5.4 LSI mathematical background

This section gives the mathematical definitions and proofs for the Latent semantic indexing method. It is based on the original articles (Deerwester *et al.*, 1990), (Berry *et al.*, 1995*c*), (Berry *et al.*, 1995*b*), (Witter and Berry, 1998).

In the heart of LSI is the term-by-document co-occurrence matrix $\mathbf{A} \in \mathbb{R}^{N \times M}$. This matrix is of the form $\mathbf{A} \equiv \{a_{ij}\}, i = 1 \ldots N, j = 1 \ldots M$. Each element a_{ij} of the matrix represents a function of the number of occurrences of term i in document j (see Example 5.1).

Id	Caption
1	"Red wine "
2	"Red Bordeaux wine platter"
3	"Red wine with blue cheese"
4	"Classic red and white (sparkling)"
5	"Great red wine being poured"
6	"Bordeaux wine platter with Swiss cheese"
7	"Red wine platter with blue cheese"
8	"Red wine platter (silhouette)"
9	"Wine and blue cheese platter"
10	"White wine and Swiss cheese platter"
11	"Red wine and pears"
12	"Great pyramid of Khufu or Cheops (red), Egypt"
13	"Middle pyramid of king Khephren, Giza, Egypt"
14	"Pyramids at Giza,"
15	"Sphinx and great pyramids, Giza, Egypt"
16	"The great sphinx, Giza, Egypt"
17	"Sun is setting on the Sphinx and great pyramids, Giza Egypt"
18	"The great Sphinx, Giza"
19	"Sphinx and great pyramid of Cheops, Giza, Egypt"
20	"Pyramid of the sun, Mexico"
21	"People viewing the pyramid of the sun, Teotihuacan, Mexico"
22	"Pyramid of the moon, red sunset, Mexico"



From this document corpus we derive the term-by-document co-occurrence matrix $\mathbf{A}_{\text{raw}}$ (below). The representation is Boolean, meaning that only the presence of the terms in the documents is represented not the number of occurrences.

Example 5.1: Constructing the raw term-by-document co-occurrence matrix

The actual function of the occurrence counts is decomposed in a product of local and global terms:

$$a_{ij} = L(i,j) \times T(i) \times D(j) \tag{5.2}$$

Where $L(i,j)$ is the local weighting of the term i in document j, $T(i)$ is the global weighting of term i across the whole collection, and $D(j)$ is the global weighting of document j with respect to the whole collection. The weighting alternatives are discussed in more detail in Section 5.4.1.

For this matrix an orthogonal base for documents and terms is then computed and coefficients chosen that give a "good" lower rank approximation of $\mathbf{A}$. By choosing a "good", but not perfect approximation we intuitively maintain the most important structure of term-document associations, and eliminate inconsistencies and ambiguities. This approach is relevant if the goal is to achieve

similarity searches. Had the application goal been discrimination among documents, these "washed out" details would have been exactly the most useful information.

In the original work, the authors chose to use the truncated singular value decomposition of $\mathbf{A}$ as the lower-rank approximation, since this is the optimal approximation method.

$$\mathbf{A} = \text{SVD}(\mathbf{A}) = \mathbf{U} \cdot \mathbf{\Sigma} \cdot \mathbf{V}^T \simeq \mathbf{U}_k \cdot \mathbf{\Sigma}_k \cdot \mathbf{V}_k^T, \tag{5.3}$$

where $\mathbf{U_k}$ and $\mathbf{V_k}$ are rank k matrices, with $k < \text{rank}(A)$.

The Appendix 5.A presents the SVD and the necessary theorems.

5.4.1 Weighting

The choice of the local and global weighting strategies in Equation (5.2) plays a significant role in the performance of the LSI technique. First lets define simple counting functions:

Definition 1 (Raw term frequency). *We define as raw term frequency the number of occurrences of a term i in document d_j with:*

$$tf(i,j) = |\{t \in d_j : t = i\}|. \tag{5.4}$$

Definition 2 (Collection-wide frequency). *We define the collection-wide frequency of a term i in document collection C with:*

$$df(i) = |\{j \in C : d_j \ni i\}| = \sum_{j \in C} tf(i,j). \tag{5.5}$$

Definition 3 (Document length). *In order to alleviate the differences between long and short documents*

we should consider the document length :

$$dl(j) = |d_j| = \sum_{i=1}^{N} tf(i,j); \tag{5.6}$$

Similarly we define the relative term frequency as :

$$rtf(j) = \frac{tf(i,j)}{dl(j)}. \tag{5.7}$$

Note that $df(i)$ is the column sum and $dl(j)$ the row sum of the raw occurrence matrix $\mathbf{L} = \{tf(i,j)\}$. Using Definitions 1 through 3 we present some of the usual local weighting strategies:

Raw occurrence count The local weight $L(i,j)$ of term i in document d_j is simply the raw term frequency:

$$L_{\text{raw}}(i,j) = tf(i,j) \tag{5.8}$$

Boolean Here a $L(i,j)$ is a simple indicator function of the presence of term i in document d_j:

$$L_{\text{bool}}(i,j) = \begin{cases} 1 & \text{if } tf(i,j) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.9}$$

Logarithmic This weighting tries to reduce large variations of frequencies in the documents. The relevance of a term to a document has been shown *not* to be linearly proportional to its frequency, a better approximation appears to be logarithmic. It is an alternative to the relative term frequency raw weighting, so the weight is given by:

$$L_{\log}(i,j) = \log(tf(i,j) + 1). \tag{5.10}$$

Augmented factor This factor attempts to equalize the relative importance of terms within a document by scaling the raw count by the number of times the most frequent term appears in the same document:

$$L_{aug}(i,j) = \frac{1}{2}\left(1 + \frac{tf(i,j)}{2 \max_i tf(i,j)}\right)$$

Unfortunately, this factor makes the term-by-document co-occurrence matrix a full matrix and is thus ignored in our investigations.

Hybrid weights These weightings use other transformations of the raw occurrence count, either by a combination of the above mentioned weights, or by some functional other than the logarithm.

Again, using the various definitions above, the second factor $T(i)$ of the weighting in Equation (5.2) could be given by :

Uniform weights Here the global weight $T(i)$ of term i is identical whatever the term i (usually 1).

$$T_u(i) \equiv 1 \quad \forall i \tag{5.11}$$

Inverse collection-wide and "idf" weights Since terms that appear very often in the collection usually have low discrimination power, yet would give high values in the co-occurrence matrix, we prefer to normalize them:

$$T_{inv}(i) = \frac{1}{df(i)} \quad \text{or} \quad T_{idf}(i) = \frac{1}{\log(df(i))}. \tag{5.12}$$

Entropy weighting Analyzing the entropy of term occurrences in a given document collection gives a good idea of the information each term carries. For instance if a term appears with a constant relative frequency in all documents than it basically carries no information, if its relative frequency varies a lot from document to document then it is more likely to carry discriminative information. Thus we define the entropy weighting by :

$$T_H(i) = H(tf(i,j)). \tag{5.13}$$

Logistic The logistic weight corresponds to the logarithm of the ratio of probabilities of a given document to contain the term or not.

$$T(i) = \log \frac{P(d \ni i)}{P(d \not\ni i)} \simeq \log\left(\frac{df(i)}{N - df(i)}\right) \tag{5.14}$$

Domain specific Here some a-priori knowledge is included with regard to global relevance of any term. For instance stop-words may be given very small weights. Or key discriminating features of the targeted domain can lend more importance to certain terms (eg. proper names in citation oriented filtering).

The final term $D(j)$ is regarded as a document normalization factor. Its role is to cope with documents of different lengths for example. The three basic functions most often used are :

No normalization

$$D_U(j) \equiv 1 \quad \forall i.$$

Length uniformization It scales all document vectors to unit length.

$$D_L(j) = 1/dl(i) \tag{5.15}$$

Global normalization This method not only scales the raw occurrences but first applies the one of the above local term weights $L(i,j)$, and one of the above described global term weights $G(i)$ and then scales the result so that each document representation has the same length.

$$D(j) = \frac{1}{\sum_i G(i)L(i,j)} \tag{5.16}$$

Notice that some similarity measures are not sensitive to document lengths like the cosine measure defined below.

The SMART taxonomy (Salton and Buckley, 1988) usually employed for summarizing the term weighting schemes is outlined in Table 5.2. It encodes the scheme by six symbols: ???.???. The first three regard the weighting applied to a query and the last three, the weight applied to the documents.

Table 5.2: The SMART weighting scheme summary.

Local weight	$t??$	Simple term frequency	$tf(i)$
	$b??$	Boolean weighting	1 iff $tf(i) > 0$
	$l??$	Logarithmic weight	$log(1 + tf(i))$
	$a??$	Augmented factor	$1/2 + \frac{tf(i,j)}{2 \max_i tf(i,j)}$
Global weight	$?n?$	None	1
	$?t?$	Inverse document frequency	$\frac{1}{\log(df(j))}$
Normalization	$??n$	None	1
	$??c$	Cosine normalization	$\frac{1}{\sum G(i)L(i,j)}$

Empirical results show, that the best strategy is logarithmic for local weights and entropy for global weights (Pecenovic, 1998). The logarithm of the occurrence count reduces large variations between documents that contain the same order of magnitude of the term. Entropy weighting gives higher weights to discriminating terms and lower weights to terms that carry little information. This result agrees with information theory. See Example 5.2 for an illustration.

The following step is to weigh $\mathbf{A}_{\text{raw}}$ according to a weighting scheme. We chose here the *???.ltn* of the SMART model (see Table 5.2). In other words the local weight is the logarithm of 1 plus the raw term frequency, the global term weight is 1 over the logarithm of the overall collection frequency. There is no normalization or global document weighting. We postpone until later the choice of the query weighting part (the *???*) in our scheme.

So with:

$$G = \log(\sum_{j=1}^{N} \mathbf{A}_{\text{raw}}(i,j)) \tag{5.17}$$

$$= (0.5139, 0.9102, 1.4427, 0.6213, 1.4427, 0.5581, 0.5139, 0.5139, 0.9102, 0.5581, 0.4551, 0.4343, 0.6213, 0.9102, 1.4427, 0.5139, 1.4427, 0.4343, 0.9102),$$

we have

$$\mathbf{A} = \log(1 + \mathbf{A}_{\text{raw}}) \cdot \mathbf{diag}(G)$$

Example 5.2: Term-by-document matrix weighting

5.4.2 Constructing the Index

The index is composed of two distinct parts that the system should keep: 1) the truncated SVD of the term-by-document matrix, and 2) all the documents projected into this lower rank approximated space.

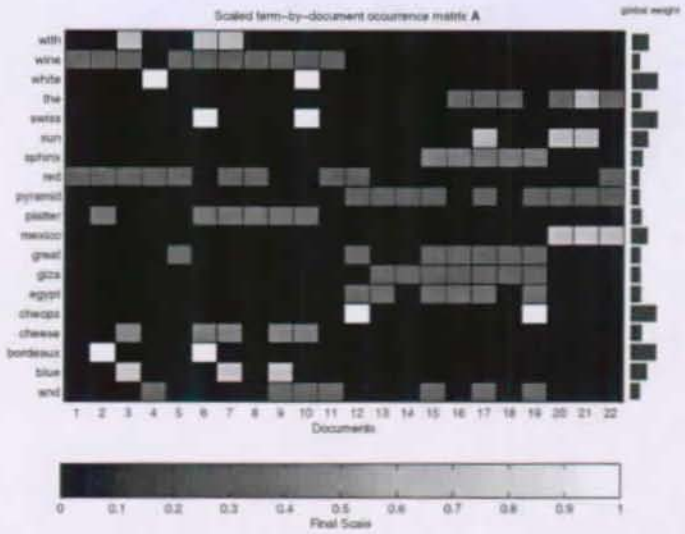
The documents that have been used to construct the index are already projected into the space, they are the rows of $\mathbf{U_k}$. It should be clear that the dimension of this space stays fairly large so that the term *index* might be misleading to the database community. Typically k would be too large for implementing point or interval access methods through R^*-trees or the like. The rank, k is usually chosen so that the norm of the difference between original matrix $\mathbf{A}$ and $\mathbf{A}_k$ does not exceed a given fraction (8% in our experiments) of the norm of $\mathbf{A}$:

$$k : \frac{||\mathbf{A} - \mathbf{A}_k||}{||\mathbf{A}||} < 0.08$$

The procedure of computing the SVD is a lengthy process for large collections, and will be done off-line. Subsequent updating strategies are introduced in Section 5.4.4.

Example 5.3 continues our presentation of the method o n the simple data-set, presenting the index in numerical form.

For layout reasons this matrix is not presented in numerical form, rather we present a graphical representation.



The term-by-document co-occurrence matrix with *???.ltn* scaling of the SMART scheme. The bar chart to the right presents the global term weights computed with the formulae of Table 5.2. The final scale is presented by the color-bar underneath the figure.

Example 5.2: Term-by-document matrix weighting (continued)

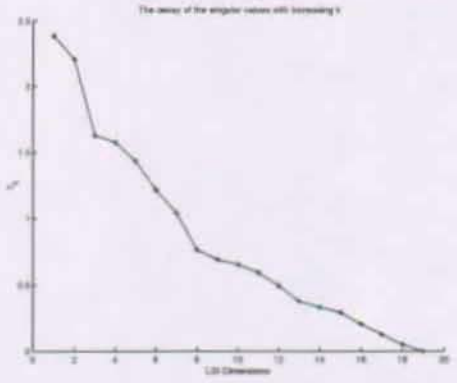
Lets continue our little example by computing the SVD of $\mathbf{A}$. The resulting decomposition is shown in Equation (5.18). Please note that $\mathbf{U_2}$ is given in transposed form for layout. Only the first two singular triplets are computed, since we are planning to project these for illustrative purposes. The illustration below shows the decay of the singular values (diagonal of Σ). In this small case the decay is not as abrupt as it can be in real life situations.

$$\mathbf{A} \simeq \mathbf{A}_2 = \mathbf{U}_2\mathbf{\Sigma}_2\mathbf{V}_2^T = \tag{5.18}$$

```
U2 =
    0.2098   -0.1693
    0.2165    0.0708
    0.3449    0.1532
    0.3363    0.1209
    0.1191   -0.3679
    0.1067   -0.3470
    0.0928   -0.3159
    0.1125   -0.3352
    0.0475   -0.1708
    0.3209    0.1155
    0.1042   -0.3635
    0.1960   -0.0295
    0.0992   -0.3308
    0.0682   -0.2502
    0.4379    0.1883
    0.0736   -0.2660
    0.2745    0.0578
    0.3133    0.0944
    0.3011    0.1250

S2 =
    2.3829         0
         0    2.2021

V2 =
    0.0643    0.0089
    0.2612    0.0987
    0.2621    0.0886
    0.1713   -0.0053
    0.0812   -0.0453
    0.5607    0.2386
    0.3142    0.1089
    0.1164    0.0292
    0.2611    0.0497
    0.4828    0.1320
    0.0957   -0.0185
    0.1227   -0.3385
    0.0460   -0.1613
    0.0277   -0.1003
    0.1111   -0.3076
    0.0770   -0.2740
    0.1403   -0.4223
    0.0697   -0.2131
    0.1611   -0.4746
    0.0554   -0.2128
    0.0619   -0.2380
    0.0821   -0.1452
```



The decay of the singular values for the 22×19 sample matrix.

Example 5.3: Building the Latent Semantic Index .

5.4.3 Querying

Once the index is in place, user queries are considered as equivalent to short documents or multi-sets of terms and represented by an $(m \times 1)$ term vector q. The reduction or projection of this vector into the lower dimensional space generated by $\mathbf{U}_k$ is performed by Equation (5.21). The process is illustrated in Example 5.4.

$$\hat{\mathbf{q}} = \mathbf{q}^{\mathbf{T}}\mathbf{U}_\mathbf{k}\mathbf{\Sigma}^{-1} \tag{5.21}$$

Then $\hat{\mathbf{q}}$ is compared to all the pre-computed reduced document vectors with an appropriate metric.

At this stage the index is ready for querying. The moment has also come to decide on the weighting scheme for queries. We will use the same scheme on both sides *ltn.ltn*, especially since our documents contain a few terms and few occurrences of multiple terms per document. In general queries will contain a single occurrence of any interesting term, thus differing greatly from the documents themselves which will contain multiples occurrences of the terms.

The raw query vector $\mathbf{q}$ is :

$$\begin{aligned}\mathbf{q}^T &= \log(1 + (0,0,0,0,0,0,0,0,0,0,0,0,0,\ 1,\ 0,0,0,0,0,\ 1,\ 0)) \odot G = \\ &= (0,0,0,0,0,0,0,0,0,0,0,0,\ 0.301,\ 0,0,0,0,0,\ 0.301,\ 0) \end{aligned} \tag{5.19}$$

where $\odot$ is the inner product, and the projection $\hat{\mathbf{q}}$ is :

$$\begin{aligned}\hat{\mathbf{q}}^T &= \mathbf{q}^T \mathbf{U}_2 \Sigma_2^{-1} = \\ &= (0.2137,\quad 0.0295)^T. \end{aligned} \tag{5.20}$$

Example 5.4: Querying the Latent Semantic Index.

The metric used by the authors is the cosine of the angle between the two vectors:

$$\text{dist}_{cos}(\mathbf{q}, \mathbf{d}) = \cos(\angle(\hat{\mathbf{q}}, \hat{\mathbf{d}})) \tag{5.22}$$

$$= \frac{\hat{\mathbf{q}}^{\mathbf{T}} \hat{\mathbf{d}}}{\|\hat{\mathbf{q}}\| \, \|\hat{\mathbf{d}}\|}. \tag{5.23}$$

Similarly a term can be considered as a multi-set of documents and represented by an $(n \times 1)$ vector $\mathbf{t}$ and be projected into the same space, generated by $\mathbf{V}_k$, by (5.24)

$$\hat{\mathbf{t}} = \mathbf{t}\, \mathbf{V_k}\, \mathbf{\Sigma^{-1}}. \tag{5.24}$$

The projection into a two dimensional space is illustrated for our toy data-set in Example 5.5.

The retrieval of relevant results is thus the search of a set of documents or terms that are within a given threshold of the query vector according to the chosen distance function. Other distance functions can be used instead of the cosine:

Euclidean The natural distance, or L_2 distance:

$$\delta_E(\mathbf{q}, \mathbf{d}) = \sqrt{\sum_{i=1} k(\hat{q}_i - \hat{d}_i)^2}. \tag{5.25}$$

Mahalanobis This distance function is given by :

$$\delta_M(q, d) = (\hat{\mathbf{q}} - \hat{\mathbf{d}})\mathbf{C^{-1}}(\hat{\mathbf{q}} - \hat{\mathbf{d}})^{\mathbf{T}}, \tag{5.26}$$

where $\mathbf{C}$ is the covariance matrix of the document representations. A specific case, usually called standardized Euclidean distance is the simplification of $\mathbf{C}$ to diagonal form containing just the variances of the data, thus assuming un-correlated dimensions.

Minkowski The Minkowski distance allows non integer powers of the exponent:

$$\delta_{Mp}(\mathbf{q}, \mathbf{d}) = \sqrt[p]{\sum_{i=1} k|\hat{q}_i - \hat{d}_i|^p}. \tag{5.27}$$

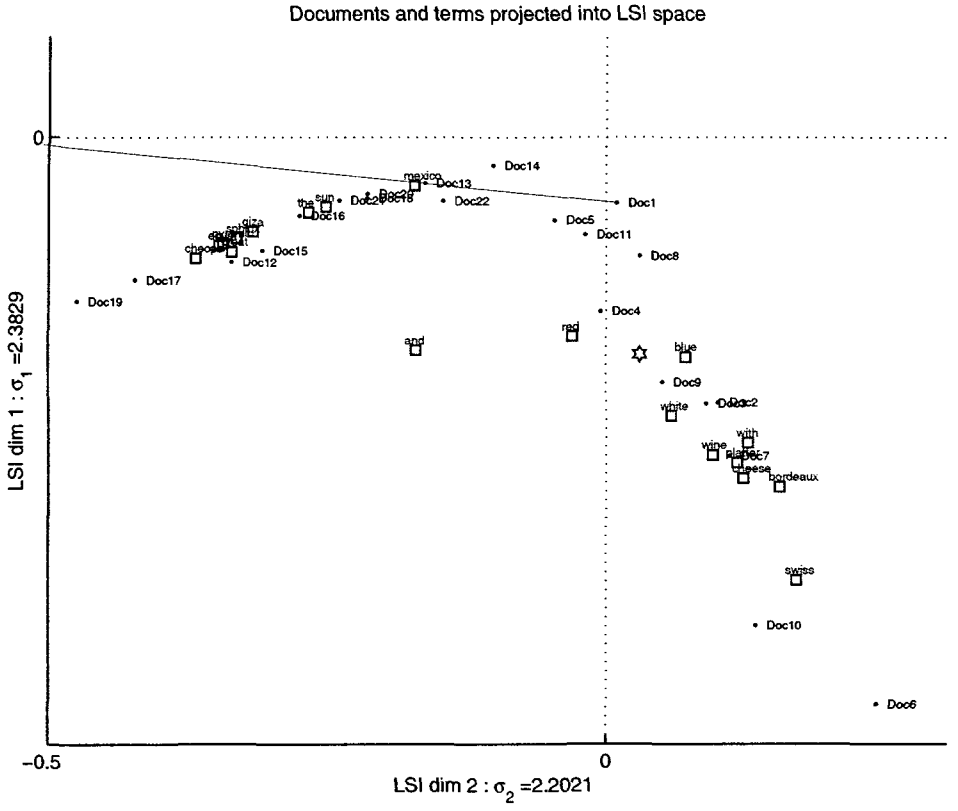
Cityblock or L1 Is the Minkowski distance with $p = 1$.

Max or L_∞ , also called the Tchebycheff distance, is the Minkowski distance with $p = \infty$, which is the equivalent:

$$\delta_\infty(\mathbf{q}, \mathbf{d}) = \max_{i=1\ldots k} |\hat{q}_i - \hat{d}_i|. \tag{5.28}$$

Two different comparison measures are illustrated in Example 5.6. In the case of the Euclidean distance, all objects located within a r radius sphere (a circle in this case) are returned. In the cosine distance, the returned entities are the ones contained in a cone around an axis pointing from the origin towards the query locations.

We can directly use the rows of the matrix $\mathbf{U}_2$ and $\mathbf{V}_2$ for projecting the data (documents and terms) into the same space. The illustration bellow presents this scenario. The large squares with the attached single labels are the terms (rows of $\mathbf{V}_2$). Likewise the documents (rows of $\mathbf{U}_2$) are represented with dots, to which is attached the concatenation of the contained terms (alphabetical term list).



The LSI space with terms and documents for the sample term-by-document co-occurrence matrix.

Notice the separation of the two themes that represent wine and pyramids, and the inter-connecting terms like "and" or "red".

Example 5.5: Viewing the produced Latent Semantic space.

5.4.4 Updating

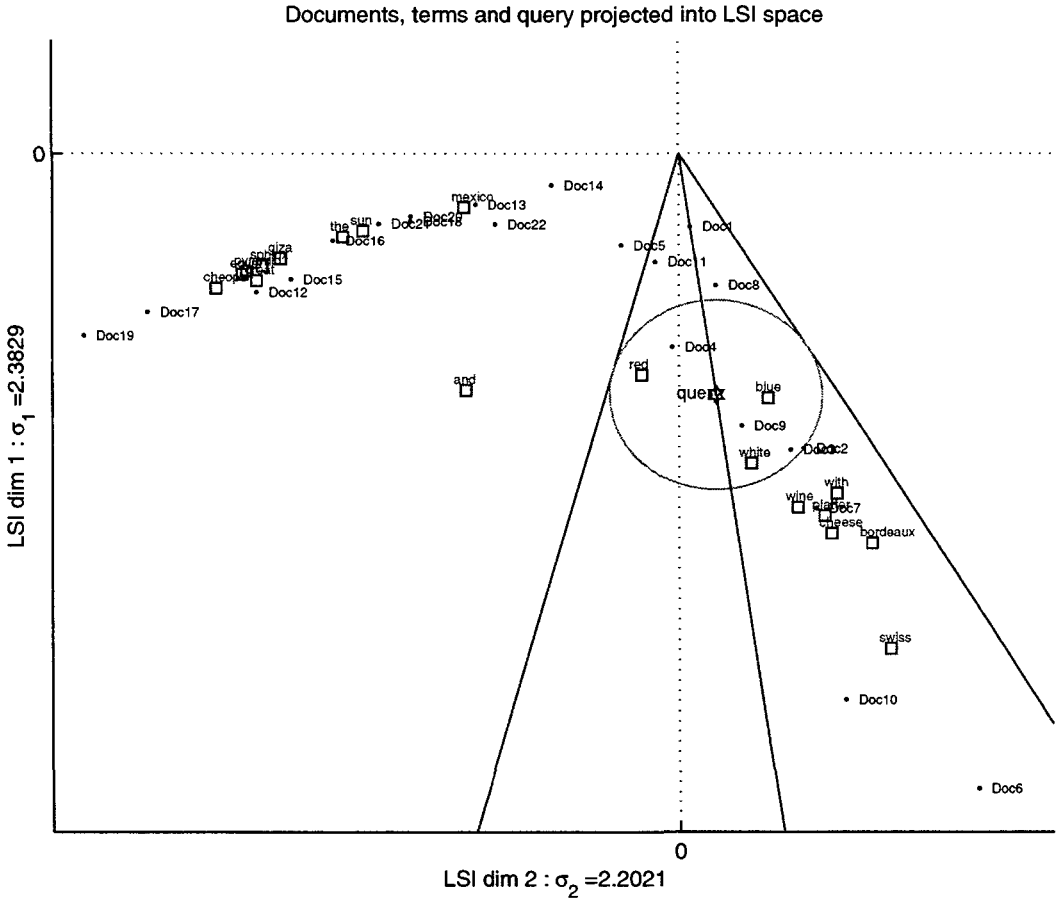
In the case where the lower-rank approximation of the matrix $\mathbf{A}$ is given by the truncated SVD the problem of computation complexity constitutes its main drawback. Adding new documents or new terms, makes the semantic model change, because new and potentially crucial information has been added. At the beginning of LSI history the only way of adding documents was to compute their reduction by Equation (5.21) and to add the resulting vectors to the columns of $\mathbf{V}_k$, and similarly with terms and columns of $\mathbf{U}_k$. By doing this the semantic model is not changed, and the weight correction if needed is not performed. This method is called by the authors "folding-in". Folding-in *does not* conserve the orthogonality of $\mathbf{U}_k$ and $\mathbf{V}_k$!

If we are adding l new documents and o new terms and we recompute the SVD of the new term-by-document matrix $\mathbf{A}^{(m+l)\times(n+o)}$, we do not update the index and the semantic model, but we recreate it from scratch. This procedure is not considered to be an updating scheme, and is hence not taken into consideration in what follows[3].

In his 1994 thesis G. O'Brian (O'Brian, 1994) exposes a scheme for updating terms, documents and weight corrections, in an elegant and efficient way. The reader should be warned that this method gives an approximation of the true new semantic model. The quality of the new indexes produced depends on the quantity of new data inserted and on the dimensionality of the current model. Periodically, the new index should be recomputed from scratch using the most recent data. The paragraphs below decompose the general updating problem into document, term or weight modification; these three steps can be carried out in any order if both terms and documents are added.

[3]Had the computation of a lower-rank approximation of $\mathbf{A}$ been a pseudo real-time operation, this approach could be considered. In the meantime, using SVD, precludes the recreation of the index.

Using Equation (5.21) we will project the query containing the keywords "red wine" and plot the results in the same projected space as presented in Example 5.5. We plot this vector in the LSI space as shown below. The last step, is to scan the document and term projections for elements that are similar. We use the cosine distance and thus the terms and documents that lie within a θ cone around the query vector are regarded as relevant. The circle plotted on the illustration bellow gives an alternative metric corresponding to the Euclidean distance or nearest neighbor. Notice that document "Doc1" which is identical to the query would be returned by a Euclidean metric much after other "more" relevant documents.



The query, represented by the star and a $\theta = 45^o$ similarity cone.

Example 5.6: Querying the Latent Semantic Index visualization

Updating documents Let $\mathbf{D}^{m\times d}$ denote the d new documents to process. $\mathbf{D}$ is sparse for the same reason $\mathbf{A}$ was. Lets append $\mathbf{D}$ to the columns of the current approximation[4] $\mathbf{A}_k$ and call this matrix $\mathbf{B} = [\mathbf{A_k}|\mathbf{D}]$. We should now efficiently compute SVD($\mathbf{B}$) $= U_\mathbf{B}\Sigma_\mathbf{B}V_\mathbf{B}^T$. Then let $\mathbf{F}$ be

$$\mathbf{F} = \mathbf{U_k^T B}\begin{bmatrix} \mathbf{V}_k & \\ & \mathbf{I}_d \end{bmatrix} = (\mathbf{\Sigma}_k|\mathbf{U_k^T D}). \tag{5.29}$$

If SVD($\mathbf{F}$) $= U_\mathbf{F}\Sigma_\mathbf{F}V_\mathbf{F}^T$ it follows that

$$\mathbf{U_B} = \mathbf{U_k U_F} \quad \text{and} \quad \mathbf{V_B} = \begin{bmatrix} \mathbf{V}_K & \\ & \mathbf{I}_d \end{bmatrix}\mathbf{V}_F, \tag{5.30}$$

since $(\mathbf{U_k U_F})^\mathbf{T}\mathbf{B}\begin{bmatrix} \mathbf{V}_K & \\ & \mathbf{I}_d \end{bmatrix}\mathbf{V}_F = \mathbf{\Sigma}_F = \mathbf{\Sigma}_B$.

Thus instead of computing the SVD of $\tilde{\mathbf{A}}^{m\times(n+d)}$ we compute the SVD of $\mathbf{F}^{k\times(k+d)}$, and two dense matrix multiplications. The process is presented in Example 5.7.

[4]Had we used $\mathbf{A}$ instead of $\mathbf{A}_k$ this would be the same as recreating the index.

The final example is the updating of the Index. Simple extension and folding in are not discussed. We present the numerics of the insertion of a new document : Doc23 : *Drinking a great red wine under the pyramids* the identified terms are : great, pyramid, red, wine. From Section 5.4.4 we extend the rows of the reconstruction $\mathbf{A}_2$ with the row of the new document thus creating $\mathbf{B}$:

$$\mathbf{B} = [\mathbf{A}_2 \mid (0,0,0,0,0,0,0,1,0,0,1,1,0,0,0,0,0,1,0)^T] \tag{5.31}$$

Then we compute $\mathbf{F}$ as defined in Equation (5.29) and its SVD($\mathbf{F}$) $= U_{\mathbf{F}}\Sigma_{\mathbf{F}}V_{\mathbf{F}}^T$:

$$\mathbf{F} = \mathbf{U_2 B}\begin{bmatrix} \mathbf{V}_2 & \\ & 1 \end{bmatrix}. \tag{5.32}$$

Finally we get a new updated index structure :

$$\mathbf{A'}_2 = \mathbf{U}_B\mathbf{\Sigma}_F\mathbf{V}_B^T = (\mathbf{U_2 U_F})\mathbf{\Sigma_F}\left(\begin{bmatrix} \mathbf{V}_2 & \\ & 1 \end{bmatrix}\mathbf{V_F}\right). \tag{5.33}$$

```
UB =                SF =                VB =
   0.2189 -0.1573      2.3942      0       0.0635   0.0127
   0.2121  0.0829           0 2.2113       0.2544   0.1139
   0.3358  0.1723                          0.2559   0.1039
   0.3290  0.1395                          0.1705   0.0051
   0.1395 -0.3606                          0.0830  -0.0402
   0.1259 -0.3405                          0.5448   0.2711
   0.1105 -0.3102                          0.3066   0.1273
   0.1311 -0.3284                          0.1142   0.0360
   0.0570 -0.1678                          0.2370   0.0640
   0.3139  0.1332                          0.4729   0.1604
   0.1233 -0.3371                          0.0960  -0.0126
   0.1973 -0.0184                          0.1392  -0.3261
   0.1176 -0.3248                          0.0530  -0.1576
   0.0821 -0.2460                          0.0327  -0.0981
   0.4278  0.1926                          0.1263  -0.2991
   0.0884 -0.2615                          0.0906  -0.2678
   0.2708  0.0728                          0.1611  -0.4114
   0.3075  0.1118                          0.0703  -0.2082
   0.2936  0.1416                          0.1845  -0.4622
                                           0.0661  -0.2083
                                           0.0737  -0.2329
                                           0.0692  -0.1406
                                           0.0992  -0.0883
```

Example 5.7: Updating the Latent Semantic Index.

Updating terms Like in the document updating procedure, lets define $\mathbf{T}^{t\times n}$ as the matrix of new terms to process and let $\mathbf{B} = \left[\frac{\mathbf{A}_k}{\mathbf{T}}\right]$. Then if $\mathbf{H}$ is defined as

$$\begin{bmatrix} U_k^T & \\ & I_t \end{bmatrix}\mathbf{H} = \mathbf{BV_k} = \left[\frac{\mathbf{\Sigma}_k}{\mathbf{TV}_k}\right],$$

and SVD($\mathbf{H}$) $= U_{\mathbf{H}}\Sigma_{\mathbf{H}}V_{\mathbf{H}}^T$. We then have

$$\mathbf{U}_B = \begin{bmatrix} \mathbf{U}_k & \\ & \mathbf{I}_t \end{bmatrix}\mathbf{U}_H \quad \text{and} \quad \mathbf{V_B} = \mathbf{V_k V_H}$$

since $\mathbf{U}_H^T\begin{bmatrix} \mathbf{U}_k^T & \\ & \mathbf{I}_t \end{bmatrix}\mathbf{BV_kV_H} = \mathbf{\Sigma_H} = \mathbf{\Sigma_B}$.

Once again instead of computing the SVD of $\mathbf{A}^{(m+t)\times n}$ we compute the SVD of $\mathbf{H}^{(k+t)\times k}$, and two dense matrix multiplications.

Weight Correction Lets suppose the weights of j terms have changed after adding a certain number of terms and/or documents. Let $\mathbf{Z}_j^{(n+d)\times j}$ represent these j weight changes, and $\mathbf{Y}_j^{(m+t)\times j}$ be a permutation matrix indicating which lines of $\mathbf{Z}_j$ apply to which lines of $\mathbf{A}_k$. We can then represent the correction step as $\mathbf{B} = \mathbf{A_k} + \mathbf{Y_j Z_j^T}$. We then have $\mathbf{Q}$ defined by the equation

$$\mathbf{Q} = \mathbf{U_k^T B V_k} = \left[\mathbf{\Sigma_k} + \mathbf{U_k^T Y_j Z_j^T V_k}\right].$$

If SVD($\mathbf{Q}$) $= U_{\mathbf{Q}}\Sigma_{\mathbf{Q}}V_{\mathbf{Q}}^T$, then it follows that

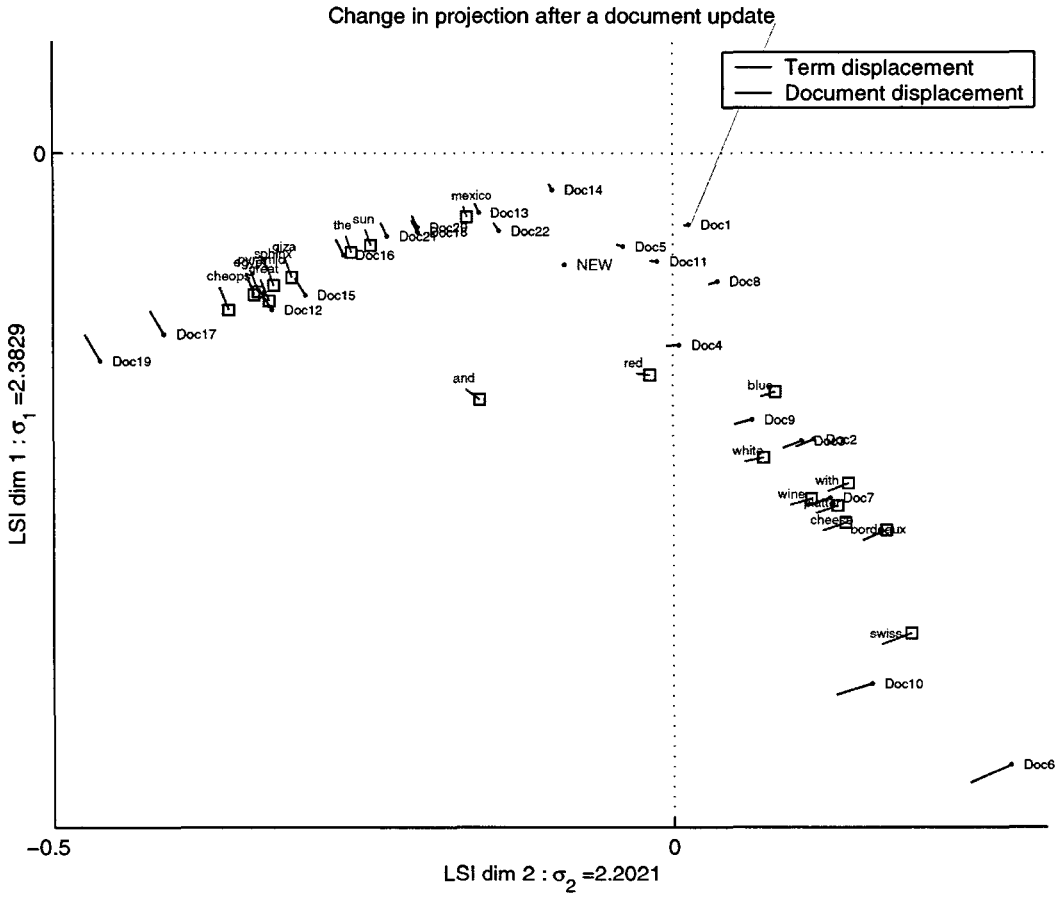
$$\mathbf{U_B} = \mathbf{U_k U_Q} \quad \text{and} \quad \mathbf{V_B} = \mathbf{V_k V_Q},$$

since $\mathbf{(U_Q U_k)^T B V_k V_Q} = \mathbf{\Sigma_Q} = \mathbf{\Sigma_B}$.

Here the computation cost is even smaller, since we compute the SVD of a $(k \times k)$ matrix and two

dense matrix multiplications instead of the SVD of an $(m+t) \times (n+d)$ matrix.

Again we do not give the expanded product, but rather we show below the behavior of the LSI generated space with respect to document and term projections.



The displacements induced in the LSI space after folding-in a new document (labeled "NEW"). The dots (documents) and squares (terms) represent the new positions whereas the starting points of the displacements represent the previous positions.

Example 5.8: Updating the Latent Semantic Index (continued)

5.4.5 Relevance Feedback

A very important topic in every information management system is relevance feedback. The user should be able to help the system retrieve useful information by using more or less sophisticated functionalities to incorporate relevant returned information in his/her query. In parallel the system should be able to learn and adapt its responses to a given query or even to a given user. This very important aspect of complex information retrieval is crucial if the data searched is not simply text.

The proposed and tested method (Deerwester *et al.*, 1990) of doing this with LSI is simply to use an average of the relevant returned documents and the query to build a new query. This gives 30% better performance in text retrieval, which is mainly due to the fact that users queries are rarely very precise and usually contain only a few words. We also apply a standard weight adaptation method proposed in (Rui *et al.*, 1998).

5.5 Implementation issues

The term-by-document co-occurrence matrix is a huge matrix. We can easily encounter collections of thousands of documents and tens or hundreds of thousands of terms. The SVD is by nature a computationally expensive operation. Luckily, the structure of the term-by-document co-occurrence matrix is very sparse (in our experiments the proportion of non-zero terms was from 0.3% to 11%). Adapted algorithms for computing either the SVD directly or computing the eigenvalue decomposition

of large sparse matrices exists. Similarly, the SVD can be solved with the SVDPACKC software bundle by Berry (1992). A description of the Lanczos iterative or block methods used in SVDPACKC can be found in (Golub and van Loan, 1983, Chap. 9), and direct implementations can be found in the Netlib repository.

In Chapter 6 we will give some more details on the issues involved when applying LSI to image/text data and on the computational software used.

5.6 Summary and discussion

This chapter was a necessary intermezzo that presented the original work on the Latent Semantic Indexing method. The justification of the method by its functionality for addressing polysemic and synonymous term usage was given. The mathematical details of the method and the consequences, especially for management procedures like updating, have been discussed. The general principles like weighting, document and term retrieval abilities of the method were exposed. The whole presentation was illustrated by a series of numerical examples on a small set of image captions. No image specific aspects were addressed, this being the scope of the following chapter.

The future research that could flow from the material presented in this chapter is principally the investigation of alternative lower-rank approximations of the term-by-document co-occurrence matrix.

5.A The Singular Value Decomposition

First let us define the singular value decomposition (SVD).

Definition 4 (SVD). *The singular value decomposition (SVD), of any matrix* $\mathbf{A}^{m\times n}$ *of rank* $r \leq q = \min(m,n)$*, denoted by* SVD(A)*, is defined as:*

$$\mathbf{A} = \mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^* \tag{5.34}$$

where $\mathbf{U}^{m\times q}$ *and* $\mathbf{V}^{q\times n}$ *are unitary matrices* ($\mathbf{U}^*\mathbf{U} = \mathbf{V}\mathbf{V}^* = \mathbf{I}_q$) *— orthogonal, if* $\mathbf{A}$ *is real —,* $\mathbf{\Sigma} = \text{diag}(\sigma_1, \cdots, \sigma_q)$ *and* $\sigma_i > 0$ *for* $i <= r$ *and* $\sigma_i = 0$ *for* $i > r$.

The first r *columns of* $\mathbf{U}$ *and* $\mathbf{V}$ *are called the left, respectively right, singular vectors and are the eigenvectors of* $\mathbf{A}\mathbf{A}^*$ *and* $\mathbf{A}^*\mathbf{A}$ *respectively. Similarly they are also the eigenvectors of*

$$\mathbf{L} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^* \end{bmatrix} \quad \textit{and} \quad \mathbf{R} = \begin{bmatrix} \mathbf{A}^* & \mathbf{0} \\ \mathbf{0} & \mathbf{A} \end{bmatrix}$$

The singular values are the non-negative square roots of the eigenvalues of $\mathbf{A}\mathbf{A}^*$ *or* $\mathbf{A}^*\mathbf{A}$.

From now on, we consider only real matrices, so the adjoint is equal to the transpose $\mathbf{A}^* = \mathbf{A}^{\mathbf{T}}$, and unitary means ortho-normal. The most interesting results and properties of the SVD for us are:

Theorem 1. *The* SVD$(\mathbf{A})$ *as defined by Def. 4 is unique except for:*

1. *Exchanging column* i *with column* j *in both* $\mathbf{U}$ *and* $\mathbf{V}$*, and exchanging* σ_i *with* σ_j.
2. *Replacing column* i *in both* $\mathbf{U}$ *and* $\mathbf{V}$ *with a unitary linear combination of columns* $l_1, l_2, \cdots, l_h$ *with the same singular value :*

$$\sigma_{l_1} = \cdots = \sigma_{l_h} = \sigma_i.$$

Proof: see (Golub and van Loan, 1983). □

Theorem 2. *Let the SVD of* $\mathbf{A}$ *be given by Def. (4) with a permutation tha guarantees that*

$$\sigma_1 > \sigma_2 > \cdots > \sigma_{r+1} = \cdots \sigma_q = 0$$

which is possible according to Thm. 1. Let, without loss of generality $m \geq n$*. Let* $R(\mathbf{A})$ *and* $N(\mathbf{A})$ *denote the range respectively the null-space of* $\mathbf{A}$*. Then*

rank property $\text{rank}(\mathbf{A}) = r$, $N(\mathbf{A}) \equiv \text{span}\{v_{r+1}, \cdots, v_n\}$ *and* $R(\mathbf{A}) \equiv \text{span}\{u_1, \cdots, u_r\}$, *where* $\mathbf{U} = [u_1 \cdots u_m]$ *and* $\mathbf{V} = [v_1 \cdots v_n]$.

dyadic decomposition $\mathbf{A} = \sum_{i=1}^{r} u_i \cdot \sigma_i \cdot v_i^T$., *meaning that only* r *singular triplets are necessary to perfectly* reconstruct $\mathbf{A}$.

norms $\|\mathbf{A}\|_F^2 = \sum_{i=1}^{r} \sigma_i^2$ *and* $\|\mathbf{A}\|_2^2 = \sigma_1^2$. *The norm* $\|\mathbf{X}\|_F$ *is the Frobenius matrix norm :*

$$\|\mathbf{X}\|_F = \sqrt{\text{trace}(\mathbf{X}^T \cdot \mathbf{X})}$$

Proof: see (Golub and van Loan, 1983). □

Theorem 2 presents several properties of the singular value decomposition of a matrix. The last two are especially interesting because they allow us to construct an approximation of $\mathbf{A}$ and estimate its quality. This is expressed by the following theorem:

Theorem 3 (Eckart and Young). *Let the SVD of* $\mathbf{A}$ *be given by Def. (4) with*

$$\sigma_1 > \sigma_2 > \cdots > \sigma_{r+1} = \cdots \sigma_q = 0$$

and define the truncated SVD approximation $\mathbf{A}_k$ *of* $\mathbf{A}$, $k \leq r$ *as*

$$\mathbf{A}_k = \sum_{i=1}^{k} u_i \cdot \sigma_i \cdot v_i^T = \mathbf{U}_k\,\mathbf{\Sigma}_k\,\mathbf{V}_k^T, \tag{5.35}$$

then

$$\min_{\text{rank}(\mathbf{M})=k} \|\mathbf{A}-\mathbf{M}\|_F^2 = \|\mathbf{A}-\mathbf{A}_k\|_F^2 = \sum_{i=k+1}^{q} \sigma_i^2$$

Proof: see (Golub and Reinsch, 1971) □

Theorem 3 states that the best rank k approximation of $\mathbf{A}$ with respect to the Frobenius norm is $\mathbf{A}_k$ as defined by (5.35). We can even say that $\mathbf{A}_k$ is the best rank k approximation of $\mathbf{A}$ with respect to any unitarily invariant norm (see (Horn and Johnson, 1991)) and hence

$$\min_{\text{rank}(\mathbf{M})=k} \|\mathbf{A}-\mathbf{M}\|_2 = \|\mathbf{A}-\mathbf{A}_k\|_2 = \sigma_{k+1}$$

No closed form solutions exist for evaluating the Singular Value Decomposition of a product or sum of matrices. Some interesting insight of the matrix stability can be found by considering the power spectrums.

Chapter 6

Retrieval method: Integrating visual and semantic cues

At the core of a multimedia retrieval system lies a set of processes that, based on the abstracted information content of the documents, create a searchable space for a set of predefined query types and interaction models. We call this set of processes, the retrieval method. Many varieties of retrieval methods exist, based on different models and assumptions, in Section 5.1 we gave an overview of the most common models and methods. We also gave a mathematical and illustrated description of the Latent Semantic Indexing model for the case of simple textual documents in the rest of Chapter 5. In Chapter 4 we described the image analysis processes that create a content description from a strictly visual point of view.

In this chapter we tie together the analysis exposed in Chapter 4 and the retrieval method of Chapter 5. We present a completely new approach to multimedia retrieval, which although based on the Latent Semantic Indexing (LSI) techniques, offers the user some additional features.

We start the presentation with a survey of related efforts for image and multimedia retrieval methods in Section 6.1. The transposition of a purely textual method to image and multimedia document collections is a true challenge. We will discuss in the Section 6.2 our investigations in defining a language for image characterization, from raw vocabulary to semantics. These constructs allow us to transparently apply LSI to our data sets. Then in Section 6.4 we expose the benefits that come from such an approach, especially concerning the interactions of visual and textual descriptors of image content. Some numerical results, based on the experiments with several data sets, are then presented in Section 6.5. For a more detailed presentation of all the performance evaluations, including those not directly linked to the retrieval method (like user satisfaction, feedback improvements and system issues), please refer to the appropriate chapters. A summary and future-research directions are presented in the closing Section 6.6. Appendix 6.B exposes a reference method to which the LSI method is compared in this chapter.

6.1 Image retrieval methods: a survey

For a more detailed survey of the current and future trends in image retrieval we refer the reader to (Smeulders *et al.*, 2000).

The problem of integrating textual and visual elements into queries for images has been addressed in several research efforts. In the WebSeek approach, Smith and Chang (1997*b*) use a model for the semantic annotation and classification methods to assign each image to a class. The queries can be executed through standard DBMS means (SQL, www interface). The two cues are however not managed in a uniform way. The QBIC project (Ashley *et al.*, 1995) has also addressed the issue in a similar way.

Other approaches link textual and visual information through probabilistic models. Vasconcelos and Lippman (1998*a*) propose a Bayesian framework for integrating textual and visual features. A similar approach is taken in (Duygulu *et al.*, 2002).

Finally the approaches proposed by Müller *et al.* (1999), use text-inspired structures for feature storage, indexing and management, thus potentially able to deal with the same problems as our own method. To our knowledge though, they do not delve into the cross-modal interactions between semantic and visual aspects of the data.

Lets also mention approaches that share the Latent Semantic Indexing method:

First La Cascia *et al.* (1998), process the caption data by LSI, but the visual characteristics are extracted into a standard vector space model. The two cues encoded as feature vectors are then combined as unrelated and indeed orthogonal sub-spaces of a single feature vector and treated by nearest neighbor search. Even with this simplistic approach the authors show a significant increase in performance over separate visual and semantic retrieval.

The second attempt at using LSI interweaving semantic with visual features is in (Westerveld *et al.*, 2000). Here, very similar ideas to ours are exposed, but apparently never implemented, tested, or evaluated. There seems to be no follow-up on this research.

Finally lets mention our own previous research: (Pecenovic, 1997, 1998; Pecenovic and Pu, 2000; Pecenovic *et al.*, 2000, 1998, 2001). In these previous approaches the global visual features were quantized using adaptive scalar quantization and mapped to occurrence counts, then added to the term-by-document co-occurrence matrix. The results as in (La Cascia *et al.*, 1998), were better than the two disjoint aspects, but the interaction of the terms was never sufficiently well defined.

6.2 Defining a language of images

In text retrieval a term is a word. The lexical word, pure alphabetic character string, can be transformed into a concept word through more or less sophisticated processing. The principal processing involves stemming (suffix and variant suppression for plural, gerund, conjugation, etc.). Additional natural language processing can also produce root words and resolve compound words. Finally a usable term is identified.

When this is transposed to images the toughest question arises:

> *What is the equivalent of a term in an image?*

Our answer to this question, simply stated, is:

> *An image term is a combination of visual and/or textual characteristics of, either the entire image, or of one of its composing regions.*

6.2.1 Defining an image term

To formalize the solution to identifying terms in an annotated image collection, lets consider an image I composed of n regions $\{r_1 \dots r_n\}, r_i \in \mathcal{R}$. In Chapter 4 we have presented a series of visual content characterizations, in the following discussion these will be used[1].

We thus associate with each region its different content models:

Shape

$$s : \mathcal{R} \rightarrow \mathcal{S} \qquad r_i \rightarrow s(r_i), \tag{6.1}$$

where $\mathcal{S}$ is the *shape* model space. Additionally, a dissimilarity function $d_S(\cdot,\cdot)$ is chosen within the model space $\mathcal{S}$.

Color

$$c : \mathcal{R} \rightarrow \mathcal{C} \qquad r_i \rightarrow c(r_i), \tag{6.2}$$

where $\mathcal{C}$ is the *color* model space. Additionally, a dissimilarity function $d_C(\cdot,\cdot)$ is chosen within the model space $\mathcal{C}$.

Texture

$$t : \mathcal{R} \rightarrow \mathcal{T} \qquad r_i \rightarrow t(r_i), \tag{6.3}$$

where $\mathcal{T}$ is the *texture* model space. Additionally, a dissimilarity function $d_T(\cdot,\cdot)$ is chosen within the model space $\mathcal{T}$.

Annotation

$$w : \mathcal{R} \rightarrow \mathcal{W} \qquad r_i \rightarrow w(r_i), \tag{6.4}$$

where $\mathcal{W}$ is the *semantic or annotation* space. Additionally, a dissimilarity function $d_W(\cdot,\cdot)$ is chosen within the model space $\mathcal{W}$.

[1] However, we prefer to highlight once again that more effective characterizations can and should be used instead. Our goal was to provide a framework and study the feasibility of the approach we propose.

A term $\mathbf{t}$ is a not completely empty tuple of the four above mentioned characteristics for each region:

$$\mathbf{t}_i = \big(s(r_i), c(r_i), t(r_i), w(r_i)\big), \text{ with } \mathbf{t} \neq (\emptyset, \emptyset, \emptyset, \emptyset). \tag{6.5}$$

We do not require that every region be characterized in all aspects, but a term must contain at least one of the characterizations. This relaxed definition leaves a great deal of flexibility for query construction. For instance, a query like

"similarly shaped cars of any color"

can be easily expressed with a term containing a null color characterization. More on the query construction aspects will be said in Section 6.4.

We also enforce, a very redundant term extraction. Each region is described using all possible combinations of the non-null terms. The redundancy is large but is very structured. By applying the LSI methods (see Chapter 5) this entails a much larger but also much sparser term-by-document co-occurrence matrix. This only amounts to a proportionally much higher decay rate of the singular values, and thus, a similar behavior for a fixed rank approximation of the matrix.

In Table 6.1 we illustrate the some of the combinations leading to term constructs and give some hints to their usage pattern. Concretely, we represent each identified region, in each indexed image, with the redundant set of terms obtained with the 15 not entirely null characterization combinations.

S	C	T	W	**Description**
•	•	•	•	A well formed term describing all visual and semantic aspects of a specific region.
•	•	•	$\emptyset$	A region of the image that the users or collection manager/indexer never annotated.
$\emptyset$	•	•	•	A term describing all the global characteristics of an image: color, texture and semantics. The whole image is not associated to any shape information.
$\emptyset$	•	•	$\emptyset$	A term describing only the global visual properties of an image. These terms can be useful for queries on purely visual and global aspects of an image, like queries for differentiating graphics and photo-realistic imagery.
•	$\emptyset$	$\emptyset$	•	An annotated shape; this type of term can arise with query by sketch, when the user has entered additional semantic information associated with a sketched object.
$\emptyset$	•	$\emptyset$	$\emptyset$	A term describing only the global color information. This is useful for color property queries for instance (see Section 7.3).
$\emptyset$	$\emptyset$	•	$\emptyset$	A term describing only the global texture information. These terms come in handy for targeted texture queries for instance (see Section 7.3).
$\emptyset$	$\emptyset$	$\emptyset$	•	A term relaying global annotation or semantics that are not associated with specific visual content, (eg. artist, atmosphere, date).

TABLE 6.1: A selection of basic term types, according to the presence or absence of some characterizing models. Most combinations have been omitted.

6.2.2 Creating a vocabulary

The next problem that we face is the creation of a vocabulary.

A large collection of documents will usually correspond to an even larger set of terms. One of the first problems faced in text-retrieval is the problem of term variation : like inflexions[2], prefix, infix or suffix word aggregation. The usual approaches to reduce these related words to a single form is stemming for the inflexions and suffixes, and dictionary-based rooting for prefixes and infixes. The latter step is frequently omitted, since these word-forms carry significant semantic differences to consider the words as different.

[2]Verb conjugation, plural forms, gender concordance, are all covered under the generic term of inflexion.

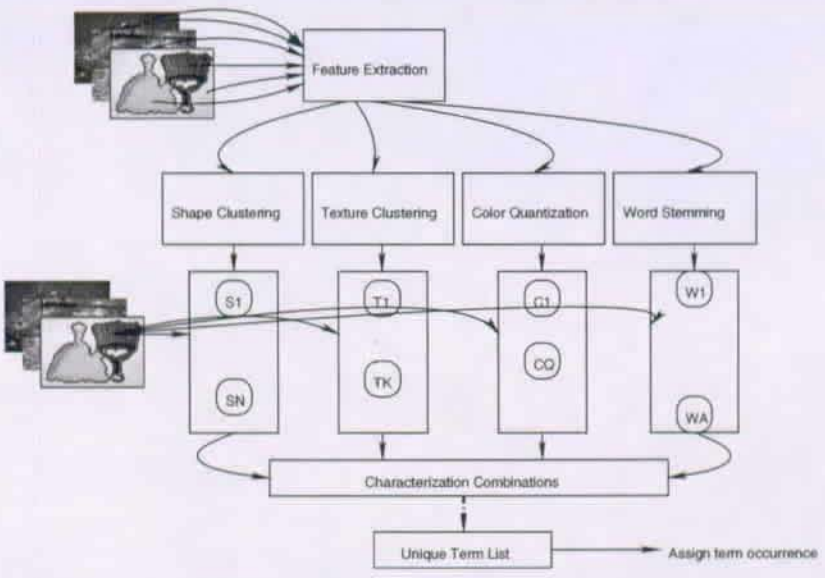


FIGURE 6.1: The finite vocabulary scenario: The processing steps taken from image content characterization to a fixed and finite vocabulary construction.

The LSI method requires the system to know, *a priori* the size of the vocabulary on which it will operate. As presented in Section 5.4.4 it is always possible to update the index with new terms, although this usually implies the re-processing of the entire collection in order to detect the presence or absence of the new term in each image. In order to avoid, or at least to minimize the necessity of term-addition updates we identified some empirical strategies to deal with most application domains and collection types:

1. The characterization can be reduced to a finite and discrete form, where each aspect can be described by one out of a finite set of values. This passage is obvious for annotation (the unique word list is the simplest example), and relatively straightforward for color characterization (color-palette). For shape and texture, however, the nature of the characterizations we propose withstands such simplification in the general setup. To tackle this issue we further consider two potentially overlapping cases:

   (a) The shapes are restricted to be in a finite set by the image collection domain, or application. This includes collections like catalogs, to some degree medical imagery, or applications where shape will never be the dominant query criterion and very rough shape description will be sufficient.

   (b) The texture characteristics will come from a texture-book, or be restricted to certain slowly varying domains. Collections and applications may include remote sensing, and again to a certain extent medical imagery.

   In both cases the discretization is dictated by the collection type and/or application domain. We end up by setting up the shape and texture "palette" or "books" by clustering the shape and texture characterizations using learning vector quantization or Self Organizing Maps (Kohonen *et al.*, 2001)[3]. See Figure 6.1 for an illustration of the process.

2. No strict discretization is performed and no limit on the number of terms is imposed *a-priori*. The terms are managed through a list of unique characterizations. These lists contain all the characterizations encountered to this point by the indexing method. Any new shape (or texture) model will be matched against this list, and if no previous term was sufficiently similar, according to the dissimilarity functions $d_S(\cdot,\cdot)$ (or $d_T(\cdot,\cdot)$), a new unique model is added. If a match was found the associated model is updated and the term corresponding to the model is judged to be present in the image.

   This scenario is more flexible and more effective, it however entails a frequent updating of the LSI index. Figure 6.2 similarly illustrates vocabulary construction in the evolving scenario.

Once a strategic choice has been made on the nature of the indexed data or the application domain (query types and patterns), a vocabulary can be constructed. In Section 6.5 we will give some results

[3]Some details on the functionality of the SOM's and other areas where we apply them are given in Section 7.4.3

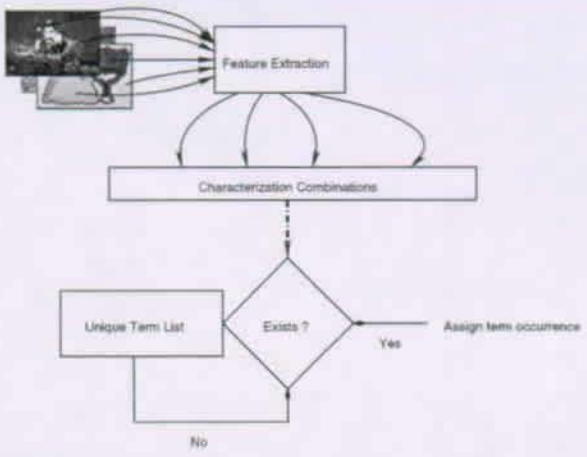


FIGURE 6.2: The evolving vocabulary scenario: The processing steps for the vocabulary creation and management.

for the impact of the size of the produced vocabularies on the performance of the method. The error rates due to the discretization steps and to the evolving nature are also explained there.

6.3 Applying Latent Semantic Indexing

After having created a vocabulary for the collection to be indexed, we must tackle the following challenge: applying the LSI. The term-by-document co-occurrence matrix as required by standard LSI contains entries that reflect the occurrence counts of each term in each document. Since all the terms that carry visual information, as defined in Section 6.2.2, are intrinsically "fuzzy", we decided to use an occurrence count that would reflect that fuzziness.

In a given document j, each region r is mapped to the set of 15 allowed characterization combinations $r_1 \dots r_{15}$. In both the fixed vocabulary and evolving vocabulary case, the K closest terms $\mathbf{i}_k, k = 1 \dots K$ are selected for each combination r_l and an occurrence proportion is assigned:

$$L(i_k, j) = e^{-k}/C, \tag{6.6}$$

where C is a normalizing factor:

$$C = \sum_{k=1}^{K} e^{-k}. \tag{6.7}$$

This process is repeated for all regions in an image. This produces a column in an raw term-by-document co-occurrence matrix. Later, when all images in the collection have been similarly processed, global weighting is applied according to Equation (5.2). The term global weighting $T(i)$ was chosen to be the entropy weighting of Equation (5.13). The document global weighting was skipped (set $\equiv 1$) since the chosen similarity function used by LSI is the cosine of Equation (5.22), which implicitly normalizes the document lengths.

Once the weighted term-by-document co-occurrence matrix has been computed, the truncated SVD is computed and the Latent Semantic Model ($\mathbf{U}_k$, $\mathbf{\Sigma}_k$, $\mathbf{V}_k$) is established.

The querying of the collection is carried out using the formulations exposed in Chapter 5, namely we can use any of the distance functions Equation (5.25) to Equation (5.28) for matching the query to the projected documents (rows of $\mathbf{U}_k$) or terms (rows of $\mathbf{V}_k$) in the the k-dimensional LSI space.

Since the basic retrieval functionality, document and term retrieval, is performed using operations on the two matrices $\mathbf{U}_k$ and $\mathbf{V}_k$, for efficiency reasons, these two full matrices are stored in column-block fashion. The column-block scheme stores the columns of the matrix in contiguous stripes on the media. The k columns of the matrix are grouped by blocks of n, and are accessed only in blocks. From the retrieval point of view, a scan on the whole matrix, with the distance computation, is performed using the first block, then only the rows that satisfy a maximum distance threshold are read from the next column-block, until all the blocks have been scanned. As a second efficiency enhancement the rows of the matrices have a permutation applied to them so as to reflect an approximate clustering based on the cosine distance to the first column block.

6.4 Emergent semantics: Relationships and interaction between visual and textual characteristics

The two different characterization domains, visual and semantic, are closely related in a well constructed collection. Retrieval based on each of the two types of characterization has been studied for some time (see Section 6.1). Obviously, some information needs are best covered by one or the other type of characterization. On the one hand, a query for paintings by Henri Matisse will probably best be answered based on the semantic meta-data associated with the images. On the other hand a query for images similar to an example painting, of which we know neither author nor title, would best be solved by visual similarity.

However, certain queries should, and can, benefit from the association of the two aspects. A very common situation is the one where the amount and quality of the annotation is not uniform for all images. A semantic query on the word "horse" would yield images that contain the word "horse", but not images that only contain the characteristic shape of a horse, its fur texture or related words ("stable", "mare", "trot"). Likewise, a visual similarity based query on an image of a horse should return all images that are annotated with the word "horse", even if the query image is not.

Latent Semantic Indexing can leverage the complex interactions that exist between visual and textual features in a rich collection of documents. Jointly, visual and semantic descriptors, and the connecting relationships, carry much more information about the content of an image than the two description types separately. We identified three basic and complementary means to make the most of this latent information. They are:

1. Visual-semantic synonymy, presented in Section 6.4.1
2. Novel query constructs, presented in Section 6.4.2
3. Image understanding and automatic annotation, presented in Section 6.4.3

6.4.1 Visual-semantic synonymy

The first step to seamless integration of visual and semantic characterizations was the term construction and identification process

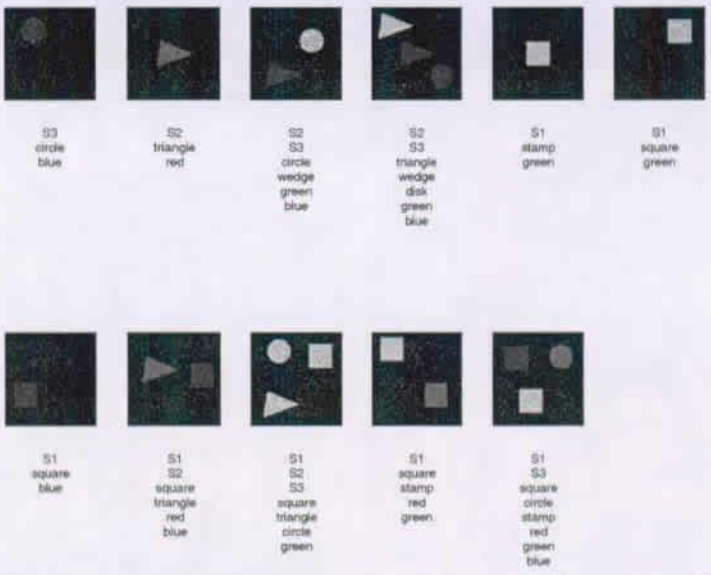
described in Section 6.2.2. The produced vocabulary contains terms that are either purely visual, purely textual, or a combination of the two. In this manner, the polymorphic information content can be effectively captured and used to best advantage.

The integration of the two aspects is already by itself a novel approach, and could be used with any retrieval method. We take a further step, one only possible, to our knowledge, with the application of the Latent Semantic Analysis method. The recurring usage patterns of related terms produces separable "directions" in the LSI projection space. For instance, if the visual color characteristic of pale blue co-occurs repeatedly with the visual shape characteristic of elongated and often straight line delimited shapes, alongside the semantic annotation "sky", a relationship of implicit synonymy will exist. LSA allows this relationship to be identified. This is how it happens:

The documents which contain the set of related terms possess a highly correlated representation in the term-by-document co-occurrence matrix $\mathbf{A}$, especially along the set of columns corresponding to those terms. After the SVD of $\mathbf{A}$, a left singular couple $(\mathbf{u}_i, \sigma_i)$ for some i will "capture" this correlation. The rows of $\mathbf{U}$, corresponding to the documents of the set, will be closer to the ideal direction (which remains precisely unknown) than other rows of the matrix. This scenario is used for retrieval of documents, based on document-like queries. We include again a simple illustration of the process in Example 6.1. Refer also to the Illustrated Examples 5.1-5.8, in Chapter 5 for a real-data, albeit, small and simple illustration.

The "added value" lies in the matrix $\mathbf{V}$. In the term-by-document co-occurrence matrix $\mathbf{A}$, each column represents a term, indicating how many times each document contains the term. A structure, similar to that of the columns mentioned above, links the terms that often appear together in the same documents. For some j, a right singular couple $(\sigma_j, \mathbf{v}_j)$ will capture this correlation. Symmetrically, the set of rows of $\mathbf{V}$ corresponding to the related terms, is more "collinear" to the ideal direction, again implicit, than the rows not relevant to the terms. This fact can be used to retrieve terms based on term-only queries.

This last functionality permeates LSI's ability to resolve synonymy. In our case the most interesting synonymy relationships are those that bind terms of different type, visual and semantic. These bindings can be used in many different ways, some of which we investigate and present in Section 6.4.2 and Section 6.4.3. The interchangeability of terms in retrieval processing offers new capabilities, like retrieving



The 11 documents contain synonymous semantic terms (e.g. "square" = "stamp"), the occurrence patterns that are captured in the term-by-document co-occurrence matrix (see Illustrated Example 6.2) highlight this synonymy through the co-occurrence of the visual features that are common (S1). Texture has been ignored in this simple example, and we use a simple vocabulary that doesn't create terms for all combinations of shape, color and semantics characterizations.

Example 6.1: Simple dataset

A =

0	0	1	0	0	1	0	0	0	0	0	1
0	1	0	0	1	0	0	0	0	1	0	0
0	1	1	0	0	1	0	1	0	0	1	1
0	2	1	0	1	0	0	1	1	0	1	2
1	0	0	0	0	0	1	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	1
1	1	0	1	1	0	0	0	0	1	0	1
1	1	1	1	1	1	0	0	0	0	3	0
2	0	0	1	0	0	1	0	0	1	1	0
2	0	1	1	0	1	1	0	0	1	1	1

We use the uniform weighting (i.e. raw occurrence counts) and the SVD of rank two is computed:

$$\mathbf{A} \simeq \mathbf{A}_2 = \mathbf{U}_2 \cdot \mathbf{\Sigma}_2 \cdot \mathbf{V}_2^T.$$

U2 =

0.1267	0.1710
0.1072	0.1362
0.2726	0.3472
0.3772	0.6486
0.1768	-0.2156
0.2011	-0.2216
0.1688	-0.1135
0.2759	0.0237
0.5214	0.0043
0.3237	-0.4647
0.4504	-0.2930

S2 =

6.3880	0
0	3.8534

V2 =

0.4529	-0.5290
0.3025	0.4693
0.2738	0.2277
0.3040	-0.2763
0.2007	0.2109
0.2147	0.0594
0.1489	-0.2529
0.1018	0.2584
0.0591	0.1683
0.1812	-0.1553
0.5271	-0.0485
0.3208	0.3718

The rows of $\mathbf{U}_2$ are the projections of the documents and the rows of $\mathbf{V}_2$ are the projections of the terms into the LSI space. The queries can be projected into this space and the cosine of the angle between the query projection and the documents(terms) can be used as similarity measure.

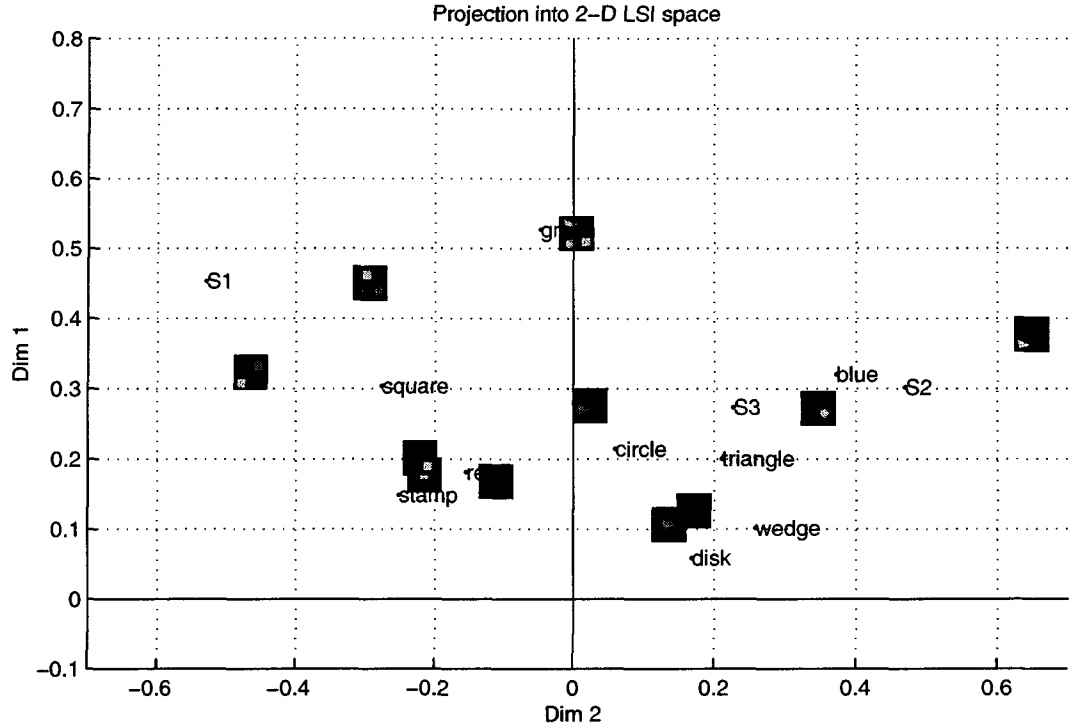
Example 6.2: The term-by-document co-occurrence matrix of the simple dataset

documents that contain a visual synonym for the word "sky" without containing the semantic annotation "sky", and vice-versa. The beauty of it all lies in the fact that this interchangeability is implicit: the system uses the closest matches for "sky", and the results will include documents containing the related visual terms[4].

If, as in certain situations, the user is not interested in lax synonymy of the above described nature, an LSI based system can perform more lexical[5] queries. This is achieved by increasing the value of k, the rank of the SVD approximation. By doing this the dimensionality of the search space increases, and the number of distinct directions increases accordingly. Pushing the limit to $k = \min(M, N)$, where

[4] Or for that matter related semantic terms like "cloud" or "sunny".

[5] Lexical in the sense that the matching is more strict, more exact.



The documents are represented as thumbnails and the terms as dots with the associated label. "S1", "S2", "S3", "red", "green" and "blue" are visual terms corresponding to three shapes and three colors, the rest are semantic terms.

The synonymy between a visual term like "S1" and the semantic terms "square" and "stamp" is easy to see if one imagines the line that connects the origin to "S1" and considers all terms that fall within a cone of a certain aperture around that line.

We see that synonymy is well exploited.

Example 6.3: The projection of the documents and terms into a 2-D LSI space.

N is the number of documents, and M the number of terms, will yield exact matching based on the term-by-document co-occurrence matrix.

6.4.2 Novel query abilities

The axes around which the Latent Semantic Retrieval method offers novel query abilities are two-fold in our approach:

1. The term retrieval ability offers new, and as of yet unexplored, cross-over query constructs.
2. The flexibility at time of query construction is much higher when using the vocabulary as exposed in Section 6.2.2.

Cross-over queries Standard information retrieval systems usually return a set of documents, or document parts, that best satisfy a query. The query is expressed as either a set of example documents, or as a description of the desired document properties (e.g. size, color proportions, contained objects). To our knowledge, no system can retrieve, based on the same queries described above, a set of relevant terms[6].

In LSI any of the following combinations is possible:

1. Document-based query → Document set results,
2. Term-based query → Document set results,
3. Document-based query → Term set results,

[6]The distinction between a term and document part is subtle, but consistent. A document part is a set of terms that usually never exists simultaneously and identically in two documents. A term is an atomic document part which usually exists in many documents.

4. Term-based query → Term set results.

This ability offers users more leverage to the information contained in a collection than simple document based queries and results.

Query expressiveness According to any one of a multitude of query paradigms[7] an LSI based system can construct a query vector to use with the same indexing structure. This not only benefits the efficiency aspect of the system, by reducing storage and development resources, but also the effectiveness of the system since the returned results can be of various types and be used in various ways to satisfy the user information need.

We anticipate here some of the discussion presented in Section 7.3 and explain how each of the query paradigms presented there can be mapped to a standard LSI query vector. All the descriptions are pertinent for queries that return documents. For the case where the system is instructed to retrieve terms, query construction is identical, it is just the query-projection matching that differs. The projection is compared to the rows of $\mathbf{V}_k$ instead of to the rows of $\mathbf{U}_k$. The first utility of such queries was explained in the preceding paragraph, it is also expanded in Section 6.4.3.

Query by properties As hinted in Section 6.2.2 document meta-data, like manipulation dates, file size or format, are stored separately, as attributes of an document. They can not directly be translated into an LSI compatible query.

Query by example The simplest query paradigm compares a query representation to the document representations of the collection. The query vector $\mathbf{q}$ is created through the same vocabulary matching as for the collection of images, and is then projected using Equation (5.21):

$$\hat{\mathbf{q}} = \mathbf{q}^T \mathbf{V}_k \mathbf{\Sigma}^{-1}.$$

The matching is performed by comparing $\hat{\mathbf{q}}$ to all rows of $\mathbf{U}_k$. The user can restrict the type of terms to be used: visual or semantic, and even more control can be given as to what sub-type of terms are to be privileged: color, texture, shape, local or global semantics, etc. These restrictions can be applied through the cancellation of unwanted terms or by a weighting according to user specified priorities.

Query by color Color presence or color proportion queries are expressed as artificial documents containing only the characterization components relevant to color. The document query vector contains non-zero elements in the columns that correspond to terms relevant to colors. The values are either proportions or indicators. The same projections are used as for a query by example above.

Query by sketch This query paradigm can be interpreted in various ways. If the sketch is a raster image, like a modified image, or a collage of image parts, the query vector is simply again analogue to the query by example. If the sketch is a point-line based sketch, the data can be analyzed and directly mapped to the most similar terms, based on the shape and color properties of the regions in the sketch. The document vector contains non-zero elements in the columns corresponding to the terms that best matched an object in the sketch. The projection is the same as above.

Query by texture This case is very similar to the color-proportion case, except that the terms used are the ones corresponding to texture.

Query by annotation An annotation query vector is simply constructed by setting the corresponding terms columns to an appropriate value. The values can be proportional to a user specified weight or identical for all terms. The same kind of projection is used as in all the above examples.

Composite queries Composite queries arise when requiring the system to retrieve documents with respect to several of the above criteria. This is obtained by applying an aggregation function either to the individual query vectors or to their projections. The aggregation function can be a numeric function, like the weighted average, or a set operation like union or intersection. The matching then proceeds with this aggregate projection as in all the above illustrated query paradigms.

[7] See Section 7.3 below to examine these from the user-interaction viewpoint.

6.4.3 Image understanding and automatic annotation

The immediate application of term retrieval as exposed above is document description. A given row $\mathbf{d}$ of $\mathbf{U}_k$ can be compared to all the rows of $\mathbf{V}_k$ and the closest matches be returned. These terms are the ones that best describe document $\mathbf{d}$. If the document didn't have any semantic annotation, the semantic terms this query returns, can be used to label the image accordingly. This process can produce automatic captioning of images. Of course the quality of this labeling can be misleading, and depends heavily on the data used for the model generation. This type of experiment was beyond our scope. More accurate feature extraction and richer data would be required for the results to be meaningful.

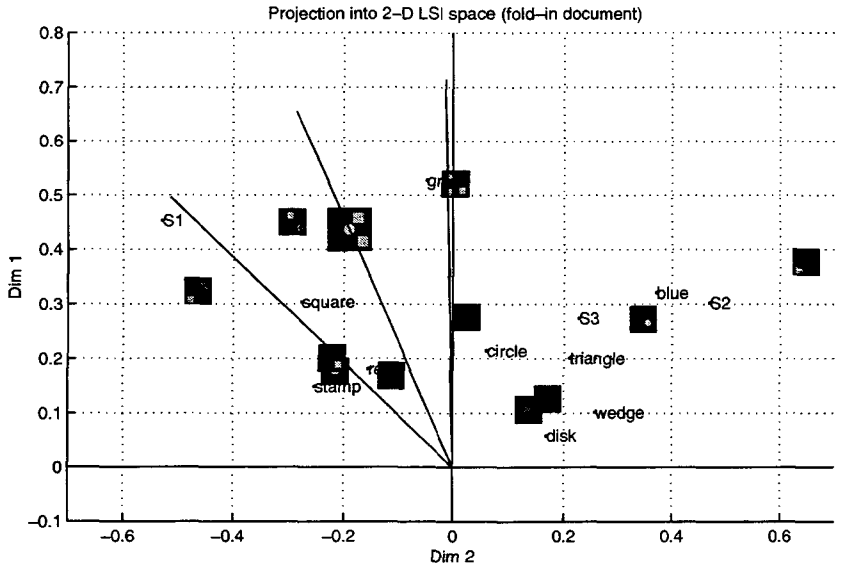
On the other hand, we have seen that a Latent Semantic Index can be used to retrieve relevant documents and terms for any query that can be projected into the approximation space: range of $\mathbf{U}_k$ for retrieving documents and range of $\mathbf{V}_k$ for retrieving terms. The various query types presented in Section 6.4.2 were all constructed using subspaces of the span of matrix $\mathbf{V}$. The natural question that comes up is : What can we obtain, and how do we interpret the results, when using the subspaces of the span of $\mathbf{U}$?

Consider the $1xN$ vector $\mathbf{t}$, the completely symmetrical construction to the query projection, using Equation (5.24):

$$\hat{\mathbf{t}} = \mathbf{t}^T \mathbf{U}_k \mathbf{\Sigma}^{-1},$$

can be considered as an "abstraction" scheme. $\mathbf{t}$ is a vector that for each document specifies whether or not it possess an abstract or unknown term. This vector can be one of the columns of the term-by-document co-occurrence matrix, but it can also be artificially constructed. The projection $\hat{\mathbf{t}}$, when compared to the rows of $\mathbf{V}_k$, can retrieve the most similar terms, i.e. terms that best qualify the set of documents corresponding to non-zero elements of $\mathbf{t}$. Thus if the user selects a set of documents, the system can answer questions like

Which terms best describe this set of documents?
or
What do these documents have in common?



We use the simplest LSI updating scheme, called folding in (see Section 5.4.4). The document vector and its projection are :

```
d =
  2   1   0   0   0   0   0   0   0   0   3   0
d * V2' * inv(S2)
ans =
    0.4368   -0.1905
```

The query based on the projection of the new document (the slightly larger image), would return for example "square" even though the label was never present in the document.

Example 6.4: The state of the Latent Semantic Indexing after adding an additional document.

Consider the simple collection of Illustrated Examples 6.1 to 6.4. We introduce an additional document into the collection the new state of the Latent Semantic Index is shown in Illustrated Example 6.4. Notice that the new document carrying only visual terms can be described effectively with the closest semantic terms ("green", "square", "stamp").

All the possibilities of the various query projections and matchings have not been equally studied, and surely offer a wealth of novel approaches. Notably for query expansion schemes, relevance feedback, search-space pruning and other enhancements

6.5 Experimental results

Finally we present here a set of performance measurements for the matter exposed in this chapter. The data we have used for performing the tests comes from several different collections. We present this data in dedicated sections Section 6.5.1 to Section 6.5.6. Along the presentation we also give the quantitative and qualitative evaluations. The discussion on the results is presented in the final section Section 6.5.7.

We consider the standard precision-recall graphs used in information retrieval. These graphs presume that ground-truth is available for a set of queries on which the statistics are collected. Additional numeric evaluators include the factor of improvement over a random system (see Appendix 6.A) and fixed-recall precision (precision evaluated at only one percentage of recall). This last measure is frequently used in text retrieval systems, and the recall percentage is usually 60%. Assessing the relevance of an image to a given information need is a much quicker process, for a human user, than assessing the same for a text document. Thus, much smaller values of precision are still acceptable. In the image retrieval setting, we prefer to use a measurement linked to the human-computer interaction, like precision after n display changes (pages, rows or other units).

We examine the evolution of the performance curves based on several aspects of each LSI construction:

Complexity The complexity C of a collection $\mathcal{X}$ is given by the product of the number of terms M and the number of documents N:

$$C(\mathcal{X}) = M \times N.$$

Intuitively, the performance should decrease with an increase in complexity, this is true, but only asymptotically. We will show that in most cases significant performance improvements occur when increasing the complexity of the index.

LSI Dimension The most important parameter, among the few that can be tuned, when applying LSI is the dimensionality k of the LSI space (the rank of the singular value decomposition) :

$$SVD(\mathbf{A})_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T.$$

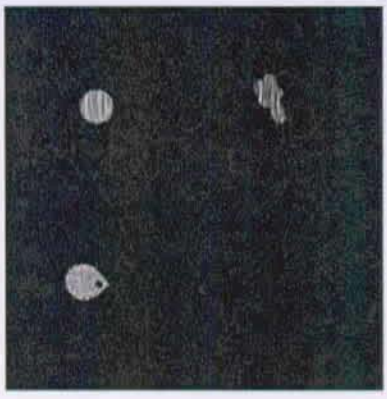
Low values of k should negatively impact performance, since the information retained is oversimplified, extreme values of k (close to $\min(M, N)$) bring LSI closer to standard vector space matching models (Story (1996)), based on the weighted occurrence counts of terms. The interesting values that lie in between bring the benefits of LSI: synonymy resolution, noise reduction and abstraction ability. Although it is difficult to argue on strict rules, empirical examination allows us to tune this parameter based on the performance of the system for certain queries. Once the SVD is computed for k all values less than k are available for experimenting. Thus, the selection of optimal k is not necessarily an additional computation burden, and can be performed regularly as new ground-truth becomes available to the system.

The next few sections present the performance evaluation on several different collections of the proposed retrieval method. The accent is on effectiveness. User-centered evaluations are covered in Chapter 7.

When the collections permit it we compare the **LSI_VS** system[8] to the reference systems (see Appendix 6.B). We examine the performance of using only semantic or only visual features, compared to an integrated approach and show that this integrated approach always out-performs both exclusive methods.

Further experiments include the behavior of term retrieval (see Section 6.4). The assessing of the term relevance is not immediate, especially since our terms map to combinations of visual and semantic

[8]The **LSI_VS** system, presented all along in this chapter, refers to the use of both visual (**V**) and semantic (**S**) characterizations of an annotated image document.



(a) A sample image.

(0,0,0, "circle"), (0,0,0, "plectra"), (0,0,0, "broccoli"), (0,0,0, "green",) (0,0,0, "marine blue"), (0,0,0, "dark cyan"), (0,0,0, "woodT"), (0,0,0, "woodT"), (0,0,0, "paperT")

(shape1, 0,0,0), (shape5, 0,0,0), (shape7, 0,0,0), (shape8, 0,0,0)

(0, color6, 0,0), (0, color7 , 0,0), (0, color8, 0,0), (0, color1, 0,0)

(0,0, texture3, 0), (0,0, texture3, 0), (0,0, texture1, 0), (0,0, texture10, 0)

(shape1, 0, texture3, 0), . . . (shape8, 0, texture10, 0)

. . .

(shape1, color6, texture3, "circle") . . . (shape8, color1, texture10, 0)

(b) Excerpt of the identified terms.

FIGURE 6.3: Sample image and terms from the *SHAPES* collection.

content. Relevant terms are those that appear in more than one document of a related set. Other definitions include a more restrictive intersection, or more lax union, of the terms as relevant set.

The final aspect we want to highlight is the synonymy resolution ability of LSI, especially for video-semantic synonymy. We would like to measure the performance of the system to answer queries on semantic keywords which have been bound to visual characteristics, and thus allow for the retrieval of images that contain the same visual characteristics, but no relevant semantic annotation.

6.5.1 SHAPES

The first data set, *SHAPES*, consists of a synthetic set of documents. We have at our disposal : A set of $S = 8$ basic geometrical shapes with their shape descriptions; A fixed color palette of $C = 8$ colors; A set of $T = 8$ pseudo-textures. A document is generated by selecting up to L shapes at random, a color and texture are chosen at random and the colored, textured shape is scaled, rotated and pasted at a random location on a uniform background. A controlled amount of Gaussian noise can be added.

For each shape, color and texture, we have at least $K = 2$ semantic labels — a true label and a set of synonym labels. The pasted objects are captioned with one of the labels at random. We can generate collections of various sizes, both in terms of the number of documents N, or the maximum and minimum number of objects per document (l and L). Any choice of shape, color, texture or label can be biased towards a particular value ($p(t)$), and especially using conditional probabilities of occurrence, ($p(t|t_o)$) given a certain characteristic already occurring in the document.

A complete vocabulary for this set can contain up to a maximum of $M = S{\cdot}C{\cdot}T{\cdot}(K{\cdot}(S{+}C{+}T)){\cdot}15$ terms. In our case, this maximum is $M = 368'640$ terms! Of course not all of these terms will be present in any reasonable collection, especially if a biasing is used for particular term co-occurrences. A unique term list is constructed "on the fly" and the actual number of terms is given with the results. Figure 6.3 presents a sample image from the set and lists an alphanumeric representation of the identified terms.

In Table 6.2 and 6.3 we present evaluations for a two hundred and three hundred document collection. The tests illustrate the results of queries on two groups of documents, the first consisting of all documents containing squares and the second all of those containing rectangles. The collection generation was not biased in any way. The words in double quotes ("") are textual query elements and those without them denote the geometric shape. The term "stamp" was used randomly as synonym for "square" and "field" for "rectangle". The final column gives the improvement ratio over a random system as defined in Appendix 6.A.

Documents : 200, Unique Terms : 320, Terms : 817 square $R_s = 27$, rectangle $R_r = 34$			
query	precision	S^a	% BTR
"square"	0.79	34	95.9
square	0.87	31	97.7
"stamp"	0.6	45	89.6
"square" & square	0.87	31	97.7
"rectangle"	0.76	45	93.4
rectangle	0.83	41	95.8
"field"	0.69	49	91
"rectangle" & rectangle	0.81	42	95.2

[a]The size S of the returned list has always been adjusted in order to achieve total recall (=1.0).

TABLE 6.2: Summary of results for *SHAPES*. A simple word like square denotes a visual shape of the square, whereas the quoted word "square" represents textual queries.

Documents : 300, Unique Terms : 484, Terms : 1381 square $R_s = 65$, rectangle $R_r = 62$			
query	precision	S	% BTR
"square"	0.72	78	90.1
square	0.84	67	95.5
"stamp"	0.71	79	90.6
"square" & square	0.86	65	96.3
"rectangle"	0.77	81	92
rectangle	0.85	73	95.4
"field"	0.72	86	89.9
"rectangle" & rectangle	0.83	75	94.5

TABLE 6.3: Summary of results for *SHAPES*. The same queries as in Table 6.2, but with larger data collection.

We notice the following facts:

- Pure visual queries outperform pure semantic queries in this simplistic setup.
- Combined visual-semantic queries however, outperform both pure variants.
- Precision is high even at extreme recall values (1.0).

These remarks can be explained by the synthetic nature of the collection and the rich annotation available for the individual regions of the images.

On Figure 6.4 we study the behavior of the same queries as the size of the collection increases. For these tests the dimensionality of the LSI index was set to half of the minimum between document and term number. The choice of this parameter will be discussed a bit more in detail in the following sections, in the case of more natural collections.

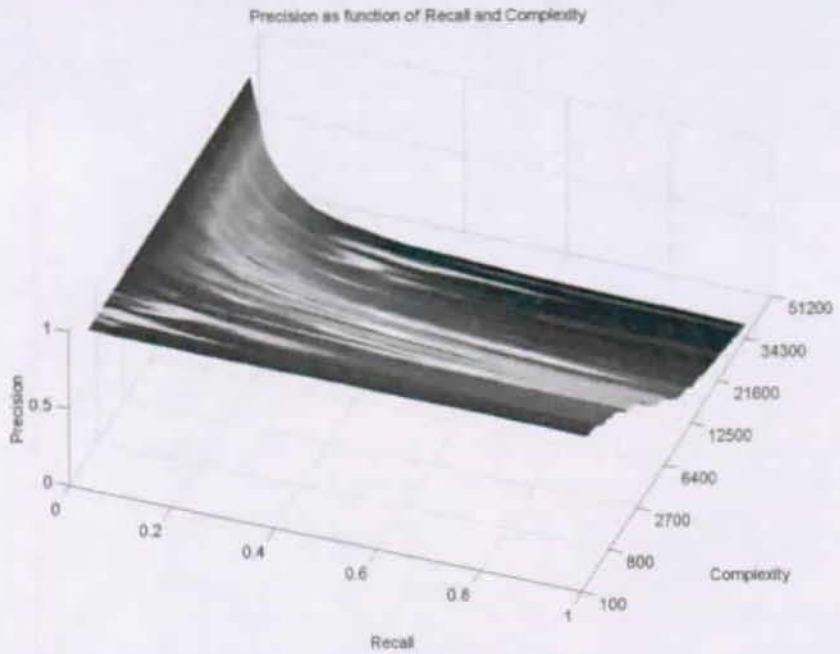


FIGURE 6.4: *SHAPES*: The precision plotted as function of recall and complexity. The different curves correspond to increasing complexity : $M \cdot N$ of the collection. Notice that along this complexity axis, the precision can actually increase slightly, although the asymptotic behavior is a rapid decay.

6.5.2 FOOD

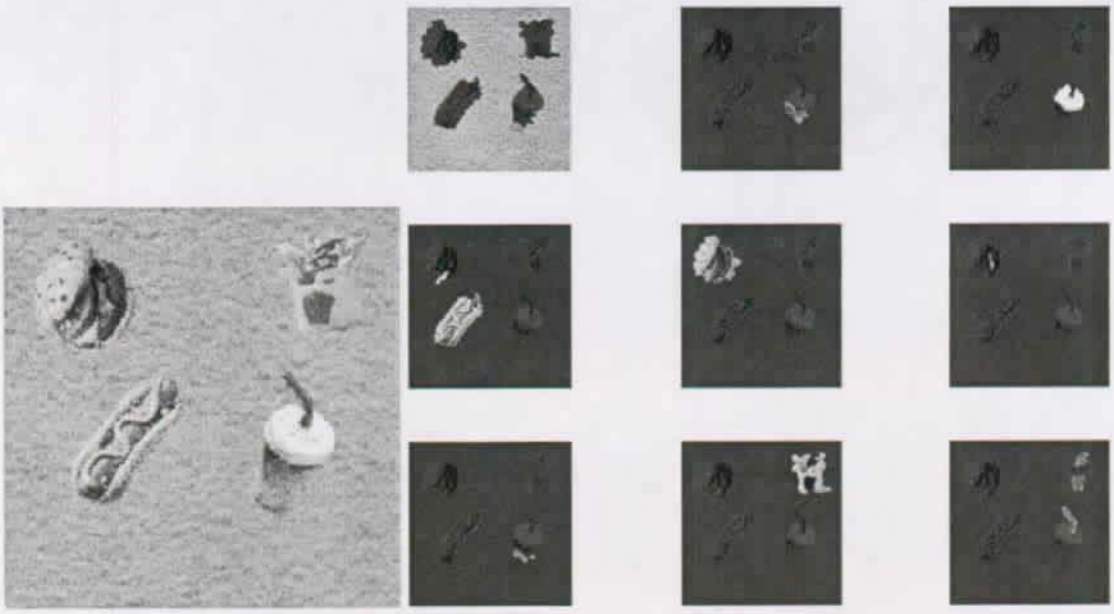
This second set *Food* is a slightly more complex synthetic database, where geometric shapes were replaced by a set of 20 drawings of snacks. Each has its proper color and texture characteristics. A random scaling, rotation and translation was applied to each randomly selected snack. A varying number of these transformed images were pasted on one of 5 textured backgrounds. The images were segmented and all characterizations presented in Chapter 4 were computed. The vocabulary was established by clustering. Shape characterization was clustered into an arbitrary number S of classes. The texture was arbitrarily clustered into a fixed number (32) of classes and the color quantized into a palette of 32 most representative colors using vector quantization. Each snack was associated with one out of four synonymous semantic labels. Since automatic one to one mapping from shapes to semantic labels was not possible —see segmentation results on Figure 6.5(b)— the semantic labeling was used only as global characterization. In other words we know the image contains a "hot-dog" but we don't know which region, or regions, are associated to this label.

The complexity of the collection can vary according to the number of distinct terms that appear, and according to the number of shape classes S. Figure 6.5 presents a sample image, its associated segmentation.

A first set of tests was conducted by generating a collection of documents where the conditional probabilities of co-occurrence between hot-dogs and burgers was set to : P(hot-dog|burger) = P(burger|hot-dog) = 0.75. In this way an artificial synonymy relation was established. Executing the queries using visual and semantic query elements, we measure the precision at recall=0.6 for the principal element and the synonym. The findings are summarized in Table 6.4.

Documents 400, Unique terms 761, Terms 1211 'burger': 14, 'hot-dog': 16, both: 11			
query	p(burger)	p(hot dog)	%BTR
"burger"	0.73	0.53	96
"hot dog"	0.45	94	0.8
burger	0.77	0.34	90
hot dog	0.39	91	0.8
"burger hot"	0.71	0.74	92
burger hot	0.79	92	0.82

TABLE 6.4: Summary of results for *FOOD*.



(a) A sample image from *FOOD* (b) The segmented regions for (a)

FIGURE 6.5: A sample image from the *FOOD* collection.

The first thing we notice is that again, pure visual queries perform better than pure semantic ones. The synonym retrieval performs well in both scenarios. In this case also the best results are obtained using both semantic and visual terms in the queries.

Figure 6.6 plots the average precision-recall curves for the visual, textual and mixed scenarios. The averages are over all the different snacks in the collection.

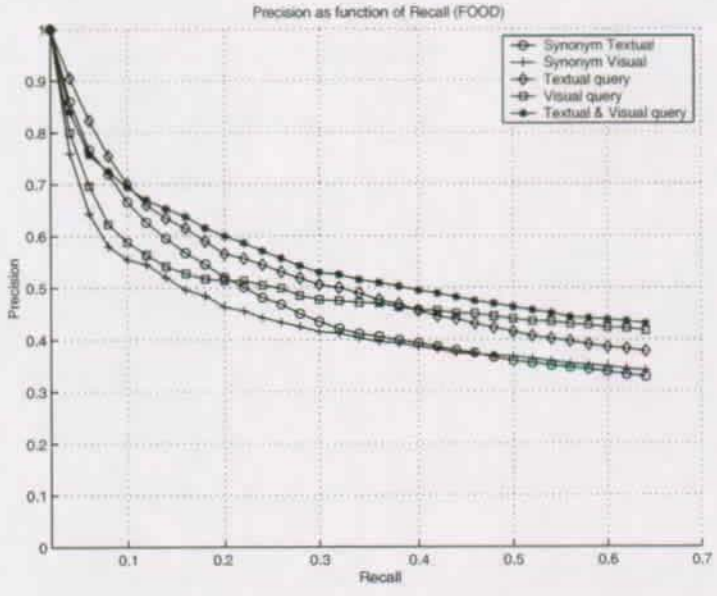
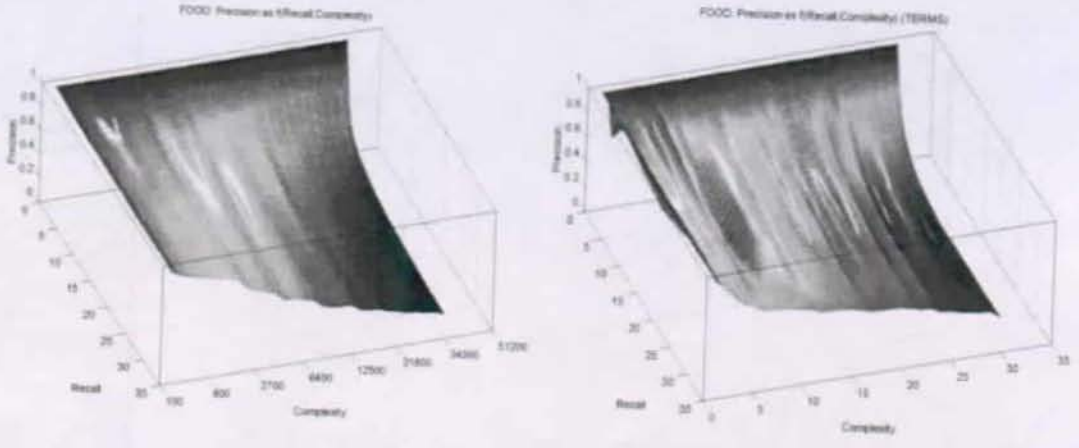


FIGURE 6.6: The precision-recall graph for the *FOOD* collection. The average precision is plotted for the query-types described above.

Figure 6.7 presents the average behavior for this collection as a function of the increasing collection complexity.



(a) Document Retrieval

(b) Term Retrieval

FIGURE 6.7: *FOOD*: The precision plotted as function of recall and complexity. The average precision, for retrieving documents (a) and terms (b), is plotted for several recall percentages. Along the opposite axis, the different curves correspond to increasing complexity: $M \cdot N$ of the collection. Notice that along this complexity axis, the precision can actually increase slightly, although the asymptotic behavior is a rapid decay.

6.5.3 COREL_F

COREL_F is a collection of photographic images with a succinct description associated to each image. It comes from the COREL Royalty free collection of CD's `www.corel.com`. It contains 6100 images, and to each is associated a global description ranging in length from 2 to 27 words. Many of these words are proper names that appear in only a single image, those that appear more than once were retained. Basic stemming was performed using an algorithmic stemmer, Porter (1980), to produce a final set of 2749 unique keywords.

The images were segmented and a fixed clustering was performed on the shape and texture characterizations into 1000 classes each. The colors were quantized to a system-wide palette of 256 colors. This would result in a maximum vocabulary size of $M \simeq 10^{13}$ terms. The actual number was a lot lower, it came to be $M = 4755$ terms. Figure 6.8 shows several images with their segmentation overlaid, and semantic annotation.

The queries we considered were based on groups of images from 6 sets of images identified during the various experiments with the previous retrieval methods studied in Pecenovic (1997) and Pecenovic (1998). The figures in parentheses refer to the number of images containing the theme: scuba-divers (100), horses (122), boats (67), Egyptian art (56), sunsets (34) and pyramids (23).

On Figure 6.9 we show the evolution of performance for various complexities of the database. The complexity (product of number of terms by number of documents) varies with the number of terms. This is proportional to each of the clustering sizes (shape and texture) and to the quantization of the color-space.

In this setting, the collection being much more varied, the performance indicators have a more constant behavior as the complexity of the index increases. The addition of new documents does not apparently influence the behavior of the system for the previously present documents.

For this collection we conducted additional comparisons with the reference systems. The **RANDOM** (see Appendix 6.A), **GFKLT** and **RFKLT** (see Appendix 6.B) were thus compared to the performance of three LSI based implementations:

LSI_VS This is the standard LSI implementation as exposed in earlier in this chapter.

LSI_S This implementation is based only on annotation terms, without any visual cues.

LSI_V This last implementation is based solely on visual terms, with no annotation.

In order to make these last three systems commensurate, we ran a first set of experiments to determine the optimal dimensionality[9]. We used these settings as starting point for further comparisons. The

[9]The optimal dimensionality is the one for which the best precision-recall curves could be attained.

FIGURE 6.8: *COREL_F*. Sample images, segmentation and annotation. The segmentation was obtained using WS-Ncut presented in Section 4.3.2. The annotation is presented in stemmed form.

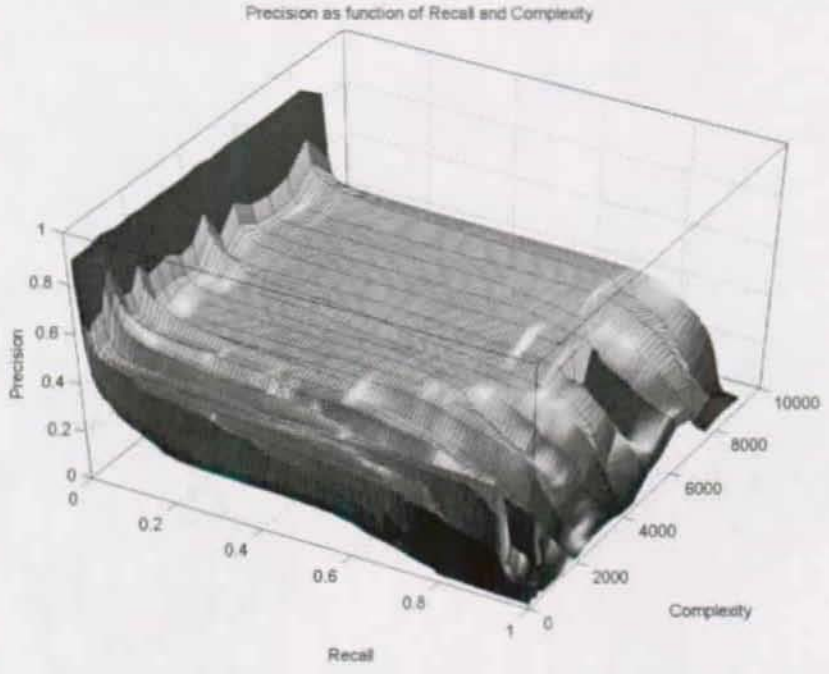


FIGURE 6.9: The precision-recall graph for the *COREL_F* collection. The average precision is plotted for the query-sets described above. The different curves reflect the complexity ($M \cdot N$ of the database).

two systems described in Appendix 6.B were also tuned by choosing the dimensionality of the wavelet-packet basis (or the number of wavelet-packet principal components) in such a way as to maximize the precision-recall curves for the query sets. On Figure 6.10 we show these comparative results.

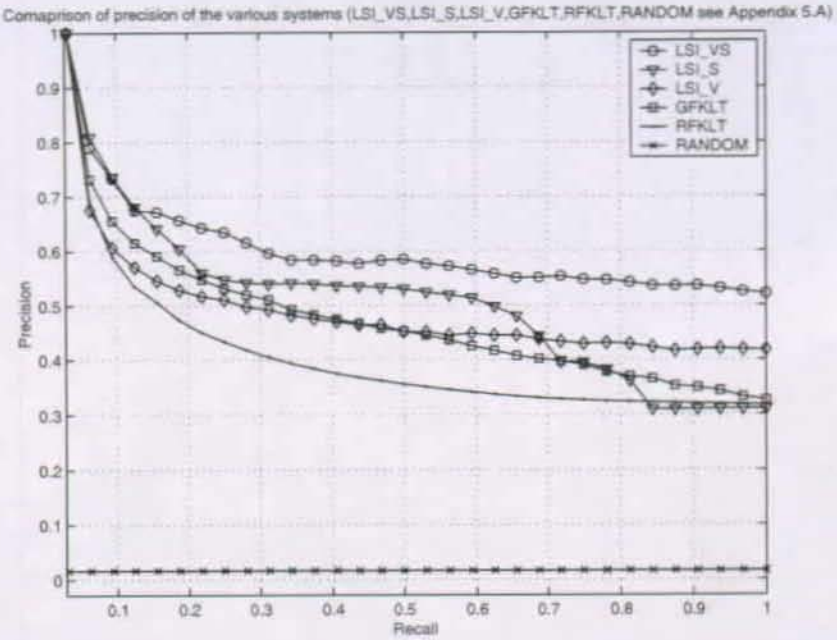
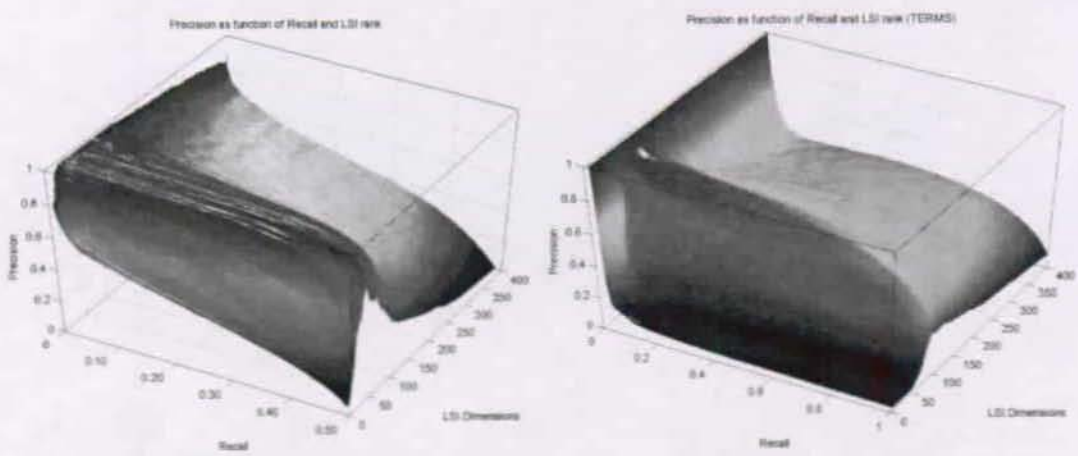


FIGURE 6.10: The precision-recall curves are plotted for the five real and the **RANDOM** system. Notice that combined visual and textual retrieval outperforms all the systems, and most importantly the two LSI-based systems: **LSI_S** and **LSI_V**. These systems exclusively use semantic respectively visual terms.

Finally in Figure 6.11 we present the results for performance according to the type of query that was used. On Figure 6.9(a) we present the results for document retrieval: L queries were issued using each of the relevant documents for a given set, the precision of the returned set of documents was averaged. The total precision was once again averaged across all groups of relevant documents described above. On Figure 6.9(b) the queries were identical, but the

returned entities were terms. If a returned term appeared in more than one of the relevant documents, the term was judged relevant. For each group of relevant images, the precision was averaged, the total precision was averaged across all groups.

The precision-recall curves were plotted as a function of the LSI dimension k (rank of the Singular Value Decomposition of the term-by-document co-occurrence matrix). An optimal choice for k seems to be between 50 and 100 dimensions for this collection. Document retrieval performs best with $k = 53$ and term retrieval with $k = 94$.



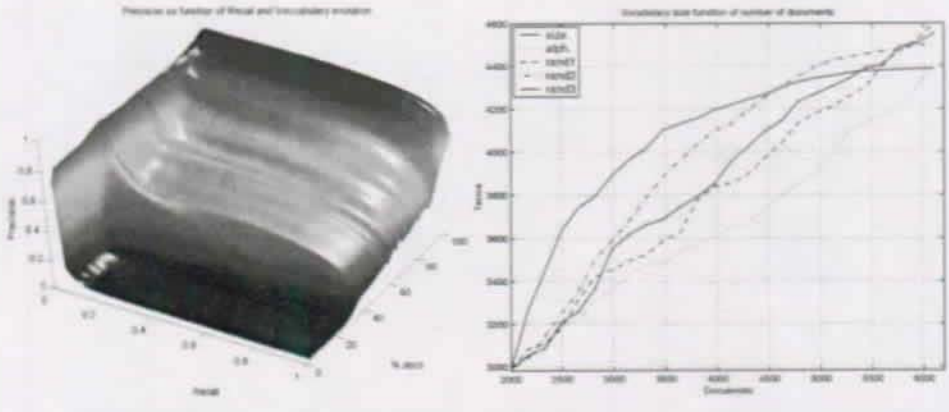
(a) Document Retrieval (b) Term Retrieval

FIGURE 6.11: *COREL_F*: The precision plotted as function of recall and rank of LSI approximation. The average precision, for retrieving documents(a) and terms(b).

6.5.4 COREL_E

The same collection was considered as for *COREL_F*, but the vocabulary was chosen using the evolving approach. The number of documents starts out with 1/3 of the total size and is increased by fixed steps (%10) of the remaining documents each time[10]. The vocabulary size did not vary much by changing the order in which the images were introduced into the collection index, but the final clustering of shape and texture characterizations did. The first vocabulary was built by introducing the documents containing more regions into the collection first. The second was obtained by inserting the documents according to their filename. The remaining three were obtained by inserting the documents randomly. Figure 6.12(a) presents this size of the vocabulary with the respect to the additional documents inserted into the index.

In the absence of a formal rule to decide which vocabulary was best, we present on Figure 6.12(b) the average performance. The queries were identical to the *COREL_F* collection and the test performed were the same. Here instead of plotting the precision-recall for various dimensions of the LSI index, we plot them according to each population-update step. The vocabulary size is plotted on Figure 6.12(a).



(a) Vocabulary size evolution

(b) Precision-recall evolution.

FIGURE 6.12: The evolution of the vocabulary size (a) and the precision-recall graph (b) for the *COREL_E* collection. Notice the steady increase of performance with each increment of the collection size.

Since the collection of documents is identical, as are the queries, we can compare the performance of the fixed vocabulary and the evolving vocabulary approaches. We selected among the five constructed evolving vocabularies the best performing one after the whole collection was ingested: random insertion 3 with size $M = 4550$. To this we compare the performance of a fixed vocabulary of approximately the same size ($M = 4755$). The results of this comparison are presented on Figure 6.13.

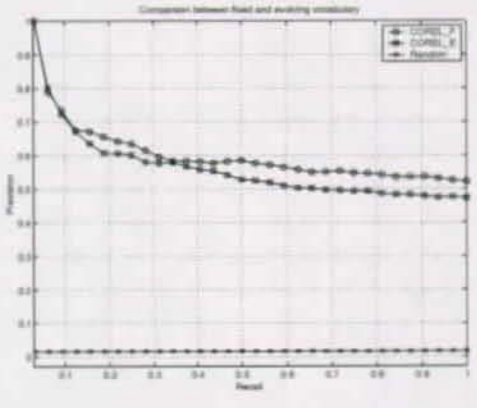


FIGURE 6.13: The precision-recall graphs for the *COREL_F* and *COREL_E* collections.

[10]The initial step includes all documents that are relevant to the test queries. An initial LSI model is computed with these 2000 documents. After each additional step, an LSI update is performed.

6.5.5 BERGER

This collection is a relatively large collection of roughly 30'000 images. It is an excerpt of the World Art Treasures collection managed by the Berger Foundation `www.bergerfoundation.ch`. The semantic labels were unevenly distributed in the collection. Some images contained up to 12 words whereas others didn't carry any annotation. Moreover, the annotations were repetitive, based mainly on the location of the shooting (country, region, town or museum) and rarely containing annotation of the content. Figure 6.14 shows a few sample images from the collection.

FIGURE 6.14: *BERGER*: Sample images and segmentation. The segmentation was obtained using WS-Ncut presented in Section 4.3.2.

The vocabulary construction was simplified. We chose to use only the characterization combinations involving a single non-null entry (shape, color, texture *or* semantics). Additionally we used the evolving approach since it exhibited only marginal deterioration in performance for the *COREL_E* collection. The final term-by-document co-occurrence matrixwas computed for $N = 27931$ documents and $M = 4612$ terms.

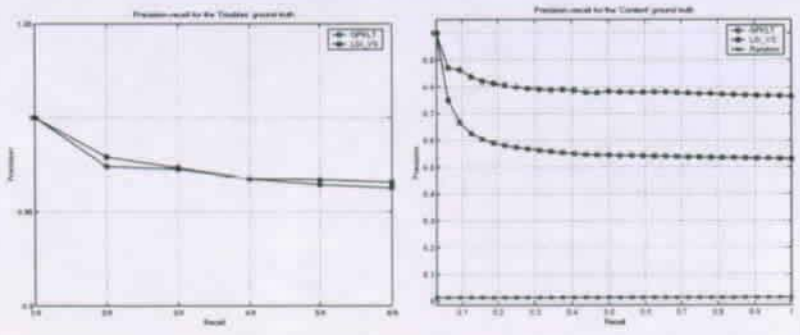
In order to identify the sets of relevant documents for performance evaluation, we considered two types of sets:

Doubles The *BERGER* collection contains many images that depict the same object or location with a small variation in viewing angle, illumination, or camera zoom. These shots appear with contiguous file-names. We identified ten such series, with at least six images each.

Content The content annotation being scarce, and the size of the collection large, it was not possible to identify images with relevant content based on manual inspection. In order to achieve this we applied the **GFKLT** method and selected a series of query images. Two un-experienced users scanned up to 500 results for each query, marking the results as relevant or not. Those images that were marked as relevant by both subjects were retained. This lead to four generic additional query sets: female portraits (56), ancient temples (141), clay sculpture (79) and bas-reliefs (127).

With this ground truth the optimal LSI dimension was found empirically to be $k = 137$ by comparing the average precisions for every k. The complexity of the collection did not vary once the whole collection was indexed. We compare the performance of the **LSI_VS**, **GFKLT** and **RANDOM** systems[11] Figure 6.15(a) shows the average precision-recall curves across the ten "doubles" groups, and Figure 6.15(b) for the content-based groups.

[11] The precision-recall graph of the **RANDOM** system is a constant equal to the ratio of relevant documents to the collection size.



(a) The "doubles" set (b) The content-based set

FIGURE 6.15: The precision-recall graph for the *BERGER* collection. On (a) we used the "doubles" and on (b) the content-based ground-truth.

6.5.6 CORBIS

CORBIS was by far the largest and most complex data-set. It contained over 100'000 images from the Corbis `www.corbis.com` digital photography collection. The images come with a rich, content descriptive and abstract annotation. After stemming and pruning proper names, the unique keyword list consisted of over 30'000 semantic terms. Thus, we decided to ignore words that didn't appear in more than 5 images, and those that appeared in more than 500. This reduced the unique word list to just over 7'000 words. Again a simplified evolving vocabulary construction was used; the resulting dictionary contained $M = 12823$ terms.

The content description was used to set-up the ground-truth, by selecting images annotated with the following keywords: horse (371), cactus (109), photograph of soccer ball (23), photograph of car (427), photograph of sunsets (534), red rose (97). Since the ground-truth was heavily biased to semantic annotation, the purely visual retrieval, gives poor precision-recall results, however the returned images were found to be "similar" by our test users.

The optimal dimension of the LSI search space was empirically computed for the above constructed ground-truth. For both **LSI_VS** and **LSI_S** it was $k = 220$ and for **LSI_V** it was only $k = 91$.

Figure 6.16 presents a sample from the test groups (along with the associated segmentation and annotation).

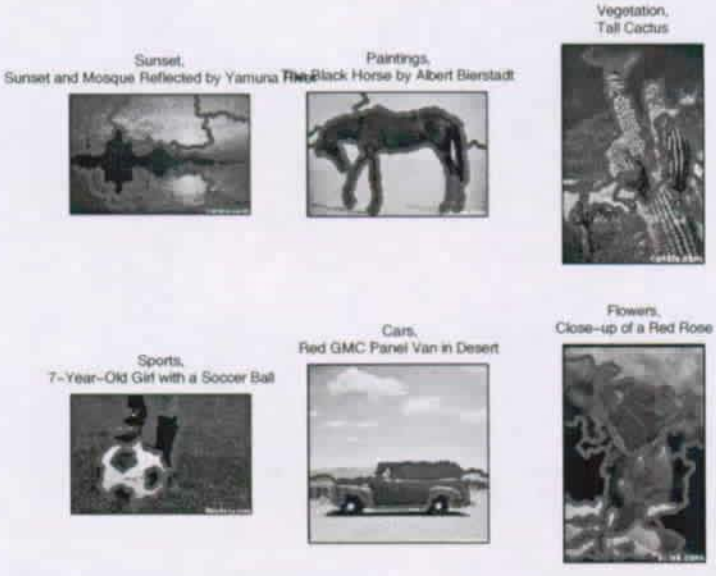


FIGURE 6.16: *CORBIS*: Sample images, segmentation and annotation. The segmentation was obtained using WS-Ncut presented in Section 4.3.2.

Figure 6.17 presents the comparison between the **LSI_VS**, **LSI_V**, **LSI_S**, **GFKLT** and **RANDOM** systems. Notice that **LSI_VS** just barely out-performs the **LSI_S** system, and both are by far the most effective retrieval methods.

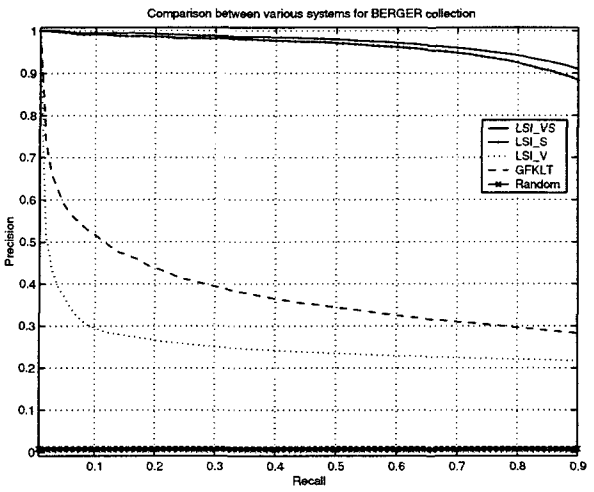


FIGURE 6.17: Comparison of the performance of various systems on the *CORBIS* collection. The average precision is plotted for the query-sets described above.

6.5.7 Result discussion

As Section 6.5.1 to Section 6.5.6 show, an increase in the complexity of the data-set does not entail a significant decrease in performance. Of course, the sets used as ground-truth data for the performance evaluation differ greatly, the *SHAPES* and *FOOD* being accurate by construction, the rest of the collections having ground-truth produced by the systems more than by human judgment. In fact, the constituting image groups have emerged during the interaction with the system or as in the case of the "doubles" sets in the *BERGER* dataset or the purely semantic groups in *CORBIS* through very tight visual respectively semantic association.

A lack of *content*-based annotation, i.e. annotation linked to specific regions of the image, in all but the synthetic sets precludes the exploitation of all the aspects identified in Section 6.4. We can however highlight the following issues that have emerged:

Synonymy resolution The LSI method was shown to be able to resolve basic notions of synonymy. Both semantic and visual synonymy could be highlighted, and more interestingly synonymy relations were uncovered by the LSI among visual and semantic content descriptors.

Performance level The precision-recall curves for the method are comparable to most current techniques based on analogous content descriptions. A higher accuracy could be achieved using more discriminating content characterizations. In a similar way performance could be increased by using content-oriented annotation. An interesting fact is that in almost all cases joint semantic and visual retrieval outperforms the mono-modal approaches.

Versatility The final remark is that the results show the feasibility of the approach of term retrieval and of its possible usefulness. More detailed testing and experiments with richer data-sets should be undertaken though to demonstrate the full capabilities.

6.6 Summary and discussion

Using the material studied in the two previous chapters, dedicated to content characterization and the Latent Semantic Indexing method, we constructed a truly novel and original approach to the integration and indexing of image+text documents. The notion of "term" for image composition was defined and discussed. Two solutions for the construction of a vocabulary from the identified terms in a document collection, necessary for implementing the LSI method, were examined. The performance of these two algorithms was measured on several data-sets.

First, the proposed method exhibits benefits from the functionality view-point. We discussed the major new functionalities of the method: synonymy resolution and term retrieval. These were illustrated on an extremely simple data-set during the presentation. They were also shown to lead to higher effectiveness than the bare document retrieval approaches. Applications of these functionalities for document summarization and result commenting was suggested, even though only basic testing was performed.

The experimental and quantitative measures of effectiveness were given based on increasingly complex data collections. In most cases a transitory increase of accuracy was measurable with the increase of complexity. The major findings regard the significant gains in effectiveness when using *both* visual and semantic terms. We proved that the LSI method systematically outperformed the reference systems.

However, for completeness sake we must consider here a question that is of major concern for many researchers: the capturing of spatial relations that may be required by the application. The retrieval of documents that depict a "red apple in a basket" must encode the positions and especially the relative positions of the identified objects. This is usually done using a spatial or conceptual graph structure of the regions. Our approach does not encode this kind of information. A major effort should be devoted to the study of how to encode this type of relationship in the term definitions or vocabulary construction.

Other important and open issues include :

- The method's new functionality should be more thoroughly studied. This is especially the case for automatic document annotation and result commenting.
- The influence on the performance of the quality of the visual content characterization as mentioned in the closing discussion of Chapter 4 should be measured.
- A study of the modifications necessary for indexing and retrieving documents of different composing media must be undertaken.
- More accurate measures could be made if a richer *content*-based annotation of the collections were available.

6.A Evaluating performance

Image retrieval research has also suffered from the lack of a unified benchmark similar to the TREC effort (Voorhees, 2001) in the text-retrieval community. Researchers have designed various algorithms for the many tasks of an image retrieval system, and claimed various "good" results, although seldom being able to compare on an even ground two approaches to a given problem. Efforts like the Benchatlon (Beretta and Marchand-Maillet, 2001) are just coming to life. It tries to offer a way of comparing different systems among each-other. Any statistic acquired through an evaluation of a system (speed, robustness, query flexibility, precision, recall) will be strongly dependent on the data that is being used and on the subjective bias of the humans choosing the "correct" answers to the test queries. However, in order to perform quantitative evaluation of a system, we need this sort of hard decisions and reliable data. The approaches to evaluation presented here rely on the availability of ground-truth data, but also on user satisfaction, interaction times and learning times, all qualitative properties.

First of all let's formalize the problem, its solution and the evaluation parameters:

Definition 5 (Elementary definitions). *Let us denote a document by d and by $\mathcal{C}$ the set of all the documents on which an information retrieval system $\mathbf{S}$ operates. A document is a set of constituting document elements denoted by $d = \{e_1^d, \ldots, e_n^d\}$. A query q is also composed of document elements or of entire documents with possible side-information like relevance or properties. We represent the result of the execution of the query q as a set of documents $\mathcal{D}$ or document elements $\mathcal{E}$ according to the type of retrieval system:*

$$\mathcal{D} = \mathbf{S}_{\text{doc}}(q, \mathcal{C}) \quad or \quad \mathcal{E} = \mathbf{S}_{\text{el}}(q, \mathcal{C}).$$

Definition 6 (Relevant set). *We denote by $\mathcal{G}^{\mathcal{C}}(q)$ the set of all documents in the collection $\mathcal{C}$ that are relevant to the query q.*

Definition 7 (Retrieved set). *We denote by $\mathcal{R}_{\mathbf{S}}^{\mathcal{C}}(q)$ the set of all relevant documents in the collection $\mathcal{C}$ that were returned by system $\mathbf{S}$.*

Definition 8 (Precision). *Precision is the proportion of documents returned that are relevant to the query:*

$$p = \frac{|\mathcal{R}_{\mathbf{S}}^{\mathcal{C}}(q)|}{|\mathbf{S}^{\mathcal{C}}(q)|}.$$

Definition 9 (Recall). *Recall is the proportion of relevant documents that were retrieved among all relevant documents in the database:*

$$r = \frac{|\mathcal{R}_{\mathbf{S}}^{\mathcal{C}}(q)|}{|\mathcal{G}^{\mathcal{C}}(q)|}.$$

We extend precision and recall with averages across a set of queries and denote, and abbreviate,

$$\bar{p} = E[p_{\mathbf{S}}^{\mathcal{C}}(\{q_1, \ldots, q_k\})] \quad \text{and} \quad \bar{r} = E[r_{\mathbf{S}}^{\mathcal{C}}(\{q_1, \ldots, q_k\})]$$

the average precision and recall respectively.

Definition 10 (The random system). *We define a random system as the system that returns a set of documents at random, with uniform probability, regardless of the processed query. By writing L the number of returned documents and K the size of the relevant set $K = |\mathcal{G}^{\mathcal{C}}(q)|$; we define X as the random variable denoting the number of retrieved documents that are relevant. With a collection of size $N = |\mathcal{C}|$, X follows a hyper-geometric distribution.*

$$X = |\mathcal{R}_{\mathbf{S}}^{\mathcal{C}}(q)|$$

$$P(X = k) = \frac{\binom{K}{N}\binom{N-K}{L-K}}{\binom{N}{L}},$$

where $\max(0, L - N + K) \leq k \leq \min(L, N)$.

The random variable $X \sim H(N, K, L)$ has expectation $E[X] = L\frac{K}{N}$, so for a return set size L the expected precision and recall are:

$$\bar{p}_{random}(L) = \frac{K}{N} \qquad \bar{r}_{random}(L) = \frac{L}{N}.$$

Definition 11 (Comparison to the random system). *For comparing a given system* **S** *with the random system we compute the average precisions at some fixed value for the recall, and for the same sizes of the returned sets* $L_i = |\mathbf{S}^{\mathcal{D}}(q_i)|$. *The ratio*

$$\rho = \frac{\bar{p}_{\mathbf{S}}}{\text{avg}(\bar{p}_{random}(L_i))}$$

equals 1 if the system **S** *is not better than random and increases as the results get more and more pertinent. We say that the system is* ρ *times more effective than the random system.*

Definition 12 (Time/space efficiency). *We define the efficiency for a given query* q *of system* **S** *by two indicators:*

$$speed(\mathbf{S}(q)) : \textit{processing time for query } q,$$

and

$$size((\mathbf{S}(q)) : \textit{size of memory necessary for executing query } q.$$

Definition 13 (Interaction count). *This user and task dependent measure counts the number of necessary queries the system must perform before the user declares being satisfied with the retrieved documents. We denote the interaction count with* $\mathbf{I}(q)$. *The same query and task is assigned to a set of different users and the average interaction count is used as system performance evaluation.*

Definition 14 (System-query complexity). *In order to grasp the performance behavior of a system* **S** *as the complexity goes up, we define the data complexity as a product of database size* N *and the query complexity* $c(q)$*:*

$$c(\mathbf{S}) = N \times c(q).$$

If the query is expressed as a set of document elements $(q) = \{e_1, \ldots, e_k\}$ *then the complexity is* $c(q) = k$, *the size of the set. If the query is a set of documents* $q = \{q_1, \ldots, q_k\}$, *the complexity is the sum of the sizes of all the document descriptions:*

$$c(q) = |\mathcal{E}(q_1)| + \cdots + |\mathcal{E}(q_k)|,$$

where the operator $\mathcal{E}(\bullet)$ *returns the document descriptors.*

Our purpose in the following chapters is to present the results and evaluations using these indicators of performance and study notably the implications of system-query complexity on the performances. A somewhat counter-intuitive result that we will present in Chapter 6 is that an increase in system complexity does not necessarily imply a decrease of system performance. In other words the performance function is not strictly monotonic, it actually, in certain situations, undergoes a temporary increase and then asymptotically decreases.

6.B Reference retrieval method

For purposes of comparison, we develop a simple retrieval method based on the raw data produced by the image content characterizations presented in Chapter 4. The details of this approach are presented in Pecenovic (1998).

The images in the collection are processed first by the watershed normalized cut method. The visual similarity among two macro-pixels p and q is given by a composition of two terms.

The first reflects color *dis*similarity:

$$c(p,q) = ||Lu^*v^*(\hat{p}) - Lu^*v^*(\hat{q})||, \tag{6.8}$$

where $\hat{p}$ is the median color of the macro-pixel p, and Lu^*v^* denotes that the color values are expressed in the Lu^*v^* color-space.

The second reflects the texture *dis*similarity:

$$t(p,q) = ||\mathbf{t}(p) - \mathbf{t}(q)||, \tag{6.9}$$

where $\mathbf{t}(p)$ is the texture description given by Equation (4.26).

Using these two *dis*similarity measures we obtain a similarity using Equation (4.9):

$$v(p,q) = e^{-\frac{\alpha c(p,q)^2 + (1-\alpha)t(p,q)^2}{\sigma_v}} \tag{6.10}$$

The maximal normalized cut value was empirically set to 0.04, the radius of the influence region r (see Equation (4.10)) was likewise set to 20% of the smaller of the two image dimensions, and the spatial normalizing factor σ_s was set to $r/2$. The final parameter α was set to $\frac{\text{var}(t)}{\text{var}(c)+\text{var}(t)}$ in order to make the color and texture dissimilarities commensurate.

Once the segmentation was computed, each region was described by the following information:

Color A normalized color histogram based on a system wide palette of 64 colors.

Texture A texture descriptor based on the moments of the significant wavelet coefficients on all subbands of three decomposition levels.

Shape A descriptor of the region's shape given by Equation (4.32).

The entire image was also characterized : i) Using the 64-color histogram and a vector quantized $u^* - v^*$ chromaticity histogram; ii) A texture representation given by Equation (4.26).

Two retrieval methods were then implemented:

Region based retrieval This system, referred to as **RFKLT** was constructed in the following way: Each extracted region was treated as an entity and its characterization was stored along with the reference to the image that the region came from. The query was constructed to reflect the same representation. The most similar regions to all those present in the query were retrieved using *dis*similarities given by Equation (4.23) and Equation (4.27). The images that these "best" regions came from were returned as the most relevant.

Global retrieval This second reference system **GFKLT** was implemented as follows : The color and texture characteristics were stored for each entire image. The query was characterized with the same model. Each image characterization was compared, as in **RANDOM**, and the lowest dissimilarity images were returned as relevant.

In both cases the extracted characterization was first pruned by removing the redundancy caused by components wit high mutual-information, than an approximate Fast Karhunen-Loève Transform was applied to reduce the dimensionality of the search space. Both of these aspects are presented in Pecenovic (1998).

In neither of the two systems did we include semantic information. In this chapter's main text a discussion is available with indications of benefits of using the integrated visual-textual approach offered by Latent Semantic analysis. The detailed performance comparison of the various systems **RFKLT**, **GFKLT** and $\mathbf{LSI}_{\mathbf{V|S}}$ can be found in Section 6.5, where the comparison is made using different image collections.

Chapter 7

User interaction: Harnessing the retrieval engine

Human computer interaction has played a significant role in information retrieval system design. As soon as interactive multimedia computing became a feasible reality, researchers started designing intuitive interfaces for multimedia document access. Specifying queries and visualizing results was a major part of these efforts. Previous research had devoted a lot of energy in designing good algorithms for image characterization and retrieval methods. But moving from a solid data representation and tested processing algorithms to a *usable* application has often proved a task of underestimated complexity. All the aspects that frighten the developers: the heavy overhead of collaborative design, user modeling and usability testing, have many times proved to be beneficial from a performance and user satisfaction point of view.

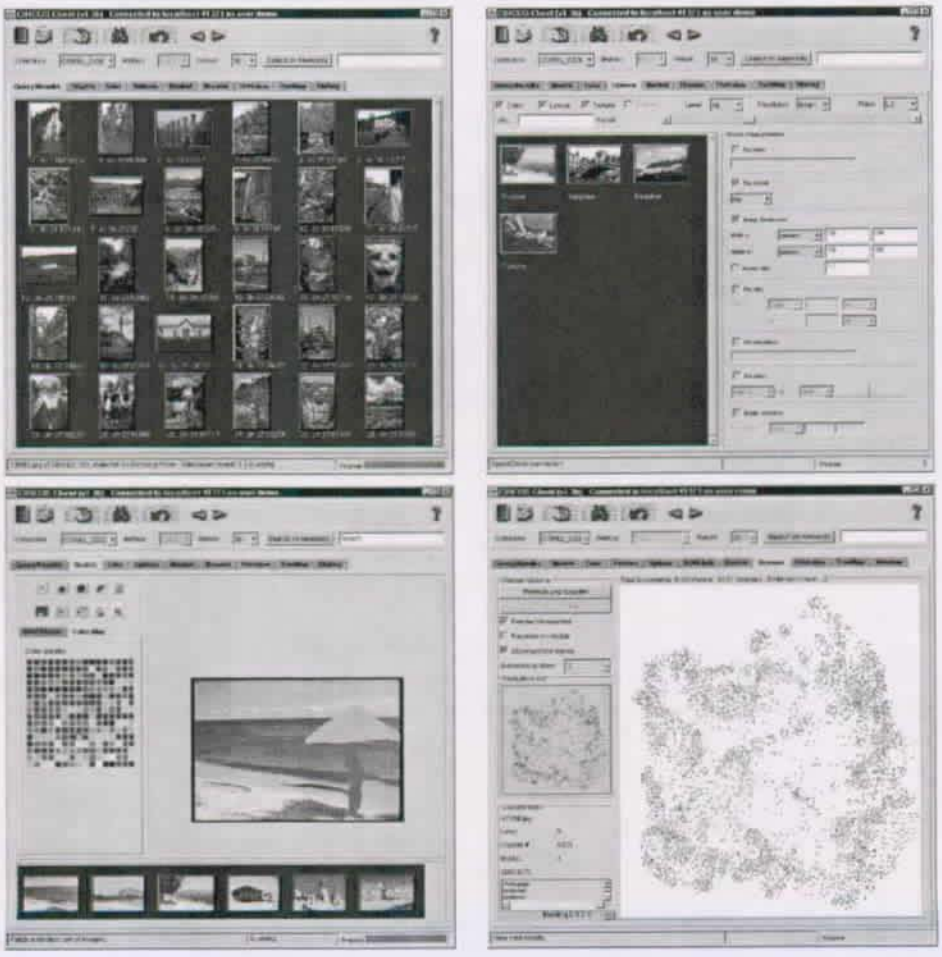


FIGURE 7.1: The principal Graphical User Interfaces of CIRCUS

In this chapter we present some of our efforts devolved to user interaction issues. The investigations being discussed are of many complementary aspects. The order of their presentation is the following:

In Section 7.1 we first try to give a short presentation of related research efforts and pointers to more detailed litterature reviews. Next we analyze the tasks that a user of an image retrieval system will normally try to carry out. The usual information needs are presented and the corresponding system functionnalities are listed. We also present in this Section 7.2 some basic system functionality that is not illustrated by the following sections.

Then Section 7.3 presents a set of query specification paradigms along with the corresponding GUIs. The focus is on functionality and much of the interaction detail is only briefly presented.

The following Section 7.4 describes our efforts to present query results to the user in meaningful ways. We investigate the basic visualizations and point out their drawbacks. Then we argue that more natural visualizations should be structured to reflect inter-result similarities and relationships. This idea of conveying the structure is then expanded upon in Section 7.5, which is dedicated to the presentation of the browsing mode of document access. It presents the interaction the user can follow to get acquainted with the overall collection contents.

Section 7.6 then explains in what way performance can be increased through tight collaboration of the user with the image retrieval system. Finally in Section 7.7, we present the conclusions, some hints for future investigations, and a list of ideas we regret not to have completed.

7.1 Human Computer Interaction for image retrieval: Related work

Classic reference related to visual information retrieval include Shneiderman (1994), Ware (2000) and a recent survey Catarci *et al.* (1997). Many useful models for interactive systems, among which the essential modes of searching for information, are illustrated in these texts. The basic axes identified in the general task of information retrieval are: direct search, browsing and navigation. Each corresponds to a set of information needs, the first is the most specific and allows access to individual documents possessing some desired characteristic or relation to other documents. Browsing fulfills the information requirements in the cases where the needs are not very precise. Navigation is accessing relevant information within a logical unit based on a spatial metaphor. The considered unit can be an entire collection or a single document.

Ware (2000) is a volume targeted at visual information systems and details many visualization techniques, some of which are related to the retrieval scenario.

Much research has been done in user modeling and "relevance feedback", both from the retrieval viewpoint (see Ruthven (2000) for a review), and from the interaction angle. Other references to the former are given in Chapter 6. The latter is treated in each of the sections below. According to the type of media concerned we also point the reader to Inder and Stader (1995) and Hearst (1999) for text retrieval, Oard (2000) for audio data and finally (Brunelli *et al.*, 1999, Section 10) explicitly for video media.

7.2 Multimedia retrieval task analysis

We first give a brief overview of the task analysis we have performed for the application of multimedia retrieval. The detailed elements like task maps can be found in Appendix 7.A. We have of course restricted most of the analysis to annotated image retrieval, but much of what follows can be applied or easily extended to other media types.

We approach the analysis first from the user side by identifying a set of basic information needs. These are precisely the requirements the system will have to cope with and try to fulfill. Then we give a list of tasks the system will be able to carry out, with in many cases alternative ways of performing the same task.

7.2.1 User information needs

When interacting with a retrieval system, the user usually has an explicit goal in mind. Even though the goal might be known, it is seldom precise. The actual information need is time varying, and more often than not the fulfillment of that need is time consuming. We have identified several needs an ideal system must provide for. We give them in decreasing order of importance. Unfortunately, it also appears to roughly follow the decreasing order of complexity for the system internals.

Happy images The user needs to find images that convey a certain abstract concept such as emotions or atmospheres, or depict a situation only interpretable by the user himself. For instance:

> "Find me some happy images." or
> "I would like to see some images that remind me of my grandmother's garden."

Help me illustrate my web page The system should find images that are related to a set of documents, which might already contain images and text. For instance given an essay on hummingbirds in Central America, the user wants to find images that would describe the natural habitat, the birds, their geographical diffusion and other "relevant" matter.

Images of rainbows The user needs to find images that depict specific objects, or vaguely specified relations among objects.

> "Find me all images with rainbows or with rainbows above forests."

Images similar to this picture This need corresponds to images that are similar to a given example or a given set of examples. Typical situations involve finding other versions of the same image in different quality or images with some resemblance to the given image. The notion of similarity here plays a crucial part, and is heavily user dependent. It also depends on the use the results will be put to.

Big images of Merida sunsets This need reflects the notion of image property, in addition to its content. The images that correspond must have for instance a minimum size and have been shot by Cristina in Merida, Mexico, in 1998.

Overview of the collection The user here is more concerned in understanding the collection of documents and what it may offer, rather than in the documents themselves. Questions like these might be examples:

> "Which collection is more likely to contain images of healthy femur x-rays at different ages?"
> "Which collection of documents more precisely describes stylistic aspects of Gothic architecture in northern Italy?"

For many of these needs, actually all but the first, we have developed effective conceptual and functional solutions. We will next describe what the actual interaction might look like for these scenarios and then present the interface elements that take part in this interaction.

7.2.2 System requirements

We first must differentiate two modes of information finding that are in a sense complementary. We consider searching as a mode of interaction where the system examines the user's requests for information, builds a representation of these and searches the collection for documents that possess similar representations. On the other tray of the scale is browsing, where the user examines the information selected and presented by the system, builds a mental representation of the system's notions of similarity and matching and then instructs it how to best adapt to the user's own viewpoint. On top of these two modes, we will build several hybrid modes where the interaction involves both of these aspects.

The system should offer query specification: based on properties of the results, on specific visual features, like color, texture, or layout, or on textual or semantic elements. Furthermore the standard queries using similarity of positive and negative samples as well as combinations of all of the above must be provided.

To exploit the converse interaction, the one allowing the user to familiarize with the system, the system should present the user with meaningful visualizations of the relationships that exist among documents, and the collection's general structure.

Some additional system functionalities, *not* described in more detail later, are briefly enumerated below.

1. The system must manage multiple collections, multiple methods of searching the collections and all the associated parameters.

2. The graphical user interface (GUI) communicates to servers that store, process, retrieve and transmit the information. It must therefore manage the network connection and associated parameters.

3. The GUI also manages a user interaction history which allows the user to backtrack through the query process and branch off in new directions.

4. Basic document management like opening the documents in a WWW browser, printing or saving of the documents.

5. Keeping a list of interesting documents through a "shopping cart" metaphor.

6. Finally it also offers basic collection management for the addition, removal and modification of documents, especially including their annotation and semantics.

7.3 Query paradigms

According to our analysis of Section 7.2, the major axes along which the performance enhancement potential must flow are query specification, and interactive visualization. No doubt the most important trait an interactive system for multimedia retrieval must demonstrate is high expressive power. This section presents our investigation into this first area, namely query construction.

CIRCUS, our multimedia retrieval system (see Section 2.3), makes available the essential queries identified by the analysis in Section 7.2.1. Restricting the domain to image and annotated image retrieval, we have identified seven major query paradigms. They are presented along with a short description of the proposed user interface. For any additional information, user manuals and the like, we refer the reader to the WWW address: `http://lcavwww.epfl.ch/CIRCUS`.

Query by properties The user is searching for all images in the collection $\mathcal{D}$ that have a certain subset of properties $q = \{q_1, \ldots, q_n\}$. The properties d_i qualifying a given document $P(d) = \{d_1, \ldots, d_k\}$ can match exactly:

$$\text{match}_{\text{exact}}(d, q) = \begin{cases} \text{true} & \text{iff } \forall j \in [1; n] \exists l \in [1; k] : d_l = q_j \\ \text{false} & \text{otherwise} \end{cases} \tag{7.1}$$

They can also be specified by ranges $q = \{[q_{11}, q_{12}], \ldots, [q_{n1}, q_{n2}]\}$ and the matching function replaced by $\text{match}_{\text{range}}$:

$$\text{match}_{\text{range}}(d, q) = \begin{cases} \text{true} & \text{iff } \forall j \in [1; n] \exists l \in [1; k] : d_l \in q_j \\ \text{false} & \text{otherwise} \end{cases} \tag{7.2}$$

Approximate matching is achieved up to threshold θ by relaxing the equality in Equation (7.1) to be an increasing monotonic function f of the absolute difference of the query and document property values:

$$\text{match}_{\text{approx}}(d, q) = \begin{cases} \text{true} & \text{iff } \sum_{j=1}^{n} f(|d_l - q_j|) \leq \theta \\ \text{false} & \text{otherwise} \end{cases} \tag{7.3}$$

where d_l has the same meaning as q_j.

The typical properties considered are meta-data like image format, presence or absence of annotation, creation dates, dimensions, resolution, and file size. The way these are to be specified depends on the application. Our solution for the property interface is presented on Figure 7.2.

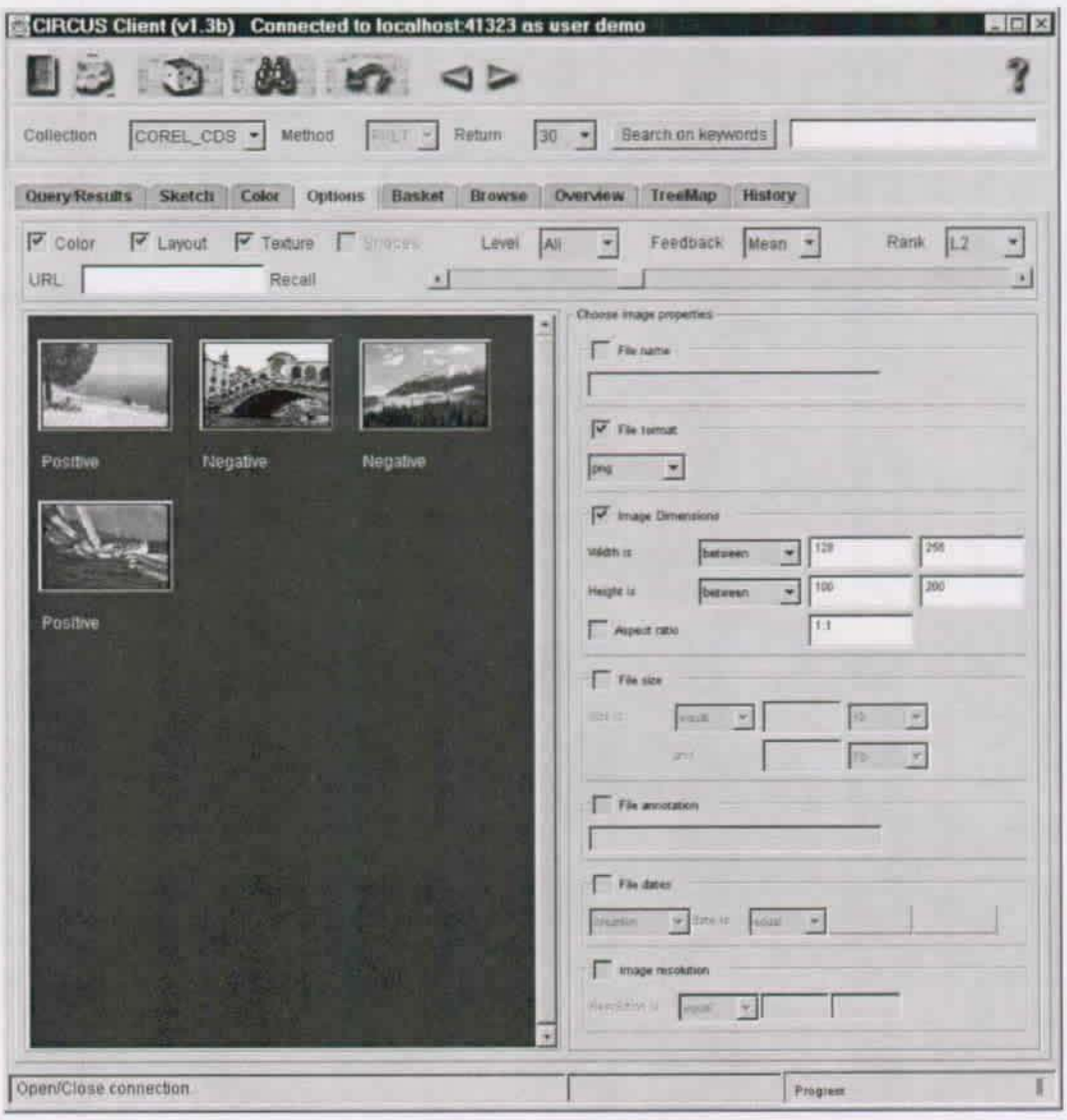


FIGURE 7.2: The query by properties interface. Along with other options for a query the user can specify the properties of the document like size and format.

Query by example The most widely used paradigm in image retrieval is query by example (QbE) Hirata and Kato (1992). The user specifies two sets of images

$$q = (\{p_1, \ldots, p_p\}, \{n_1, \ldots, n_n\}) \tag{7.4}$$

the first which are vaguely similar to the target and the second that do not correspond to her/his expectations. The system returns a set of images most similar to the positive examples p_i and at the same time least similar to the negative ones n_j. Again, according to which examples are selected the solution can vary greatly in terms of visible system behavior.

$$\text{match}_{\text{QbE}}(I, q) = \begin{cases} \text{true} & \text{iff } \sum_{i=1}^{p} d(I, p_i) \leq \theta_P \text{ and } \sum_{i=1}^{n} d(I, n_i) \geq \theta_N \\ \text{false} & \text{otherwise} \end{cases} \tag{7.5}$$

where $d(\cdot, \cdot)$ is a system dependent dissimilarity evaluation function, and θ_P and θ_N are two thresholds. In Section 4.4 one can find more detailed discussion of the dissimilarity function. In Figure 7.3 we show a sample implementation of how the CIRCUS system allows for multiple example selection.

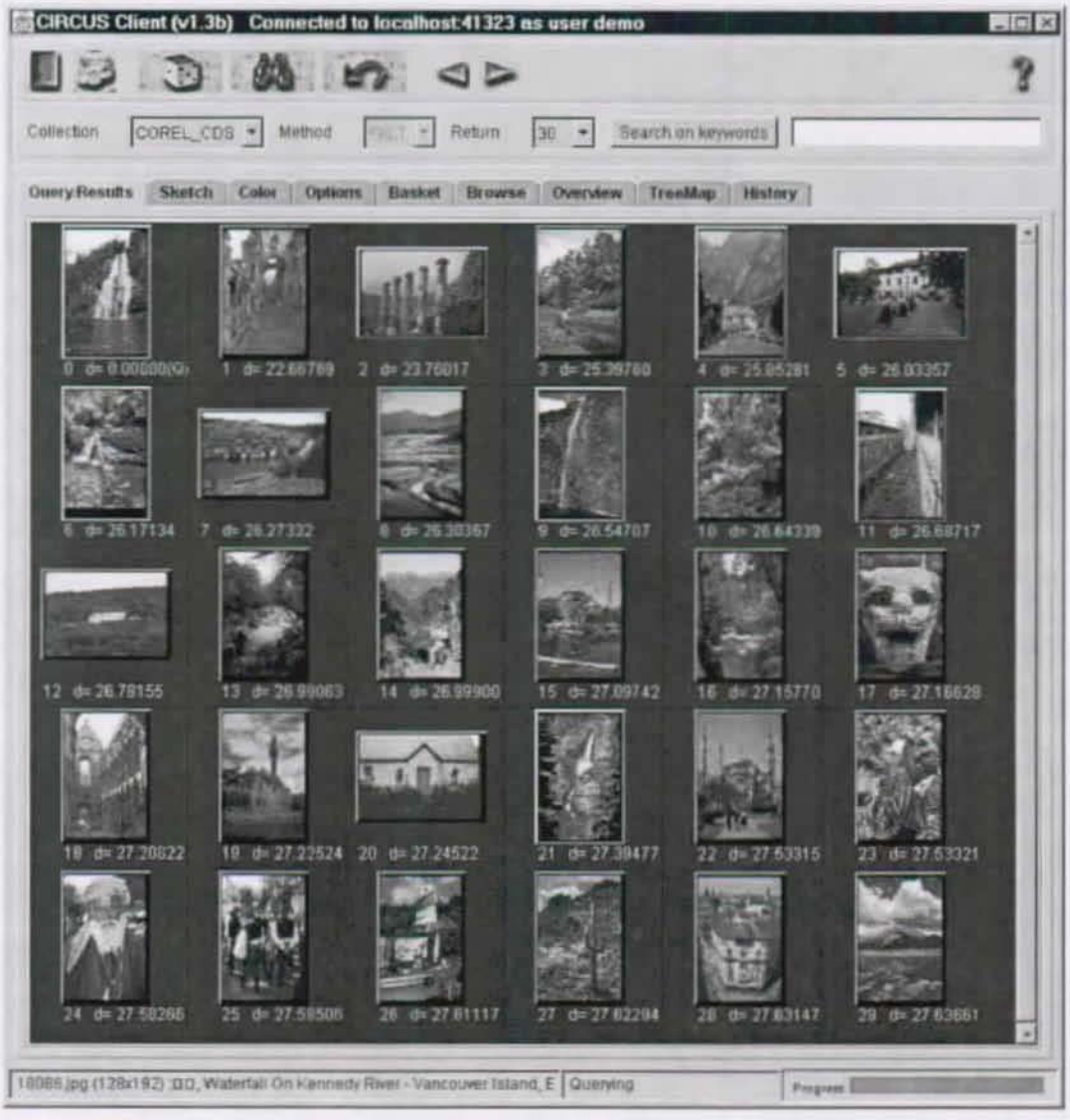


FIGURE 7.3: The query by example paradigm. The user has selected a few relevant images by mouse clicks (green border) and a few negative examples by selecting the appropriate option in the context-menu (red border).

Query by color In order to specify information needs in terms of color content, the user must be able to specify the proportions of colors she/he would like to see in the result images.

$$q = \{(c_1, p_1), \ldots, (c_n, p_n)\} \tag{7.6}$$

$$\text{match}_{\text{QbC}}(q, I) = \begin{cases} \text{true} & \text{iff} \quad \forall k \in [1; n] f(|CP(I, c_k) - p_k|) \leq \theta \\ \text{false} & \text{otherwise} \end{cases} \tag{7.7}$$

where $CP(d, c)$ is the proportion of color c in image d, $f : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ a monotonic function of the absolute difference in color proportion. As previously θ is maximum difference threshold. Eventually, it would be easier for a user to pick the colors either from a standardized palette or from sample images she/he already has seen. We propose an interface that does just that, illustrated on Figure 7.4. In the lower part of the screen a set of tentative results can be displayed interactively, as the user modifies the query. An automatic color selection method is also provided as well as the threshold value (vertical slider).

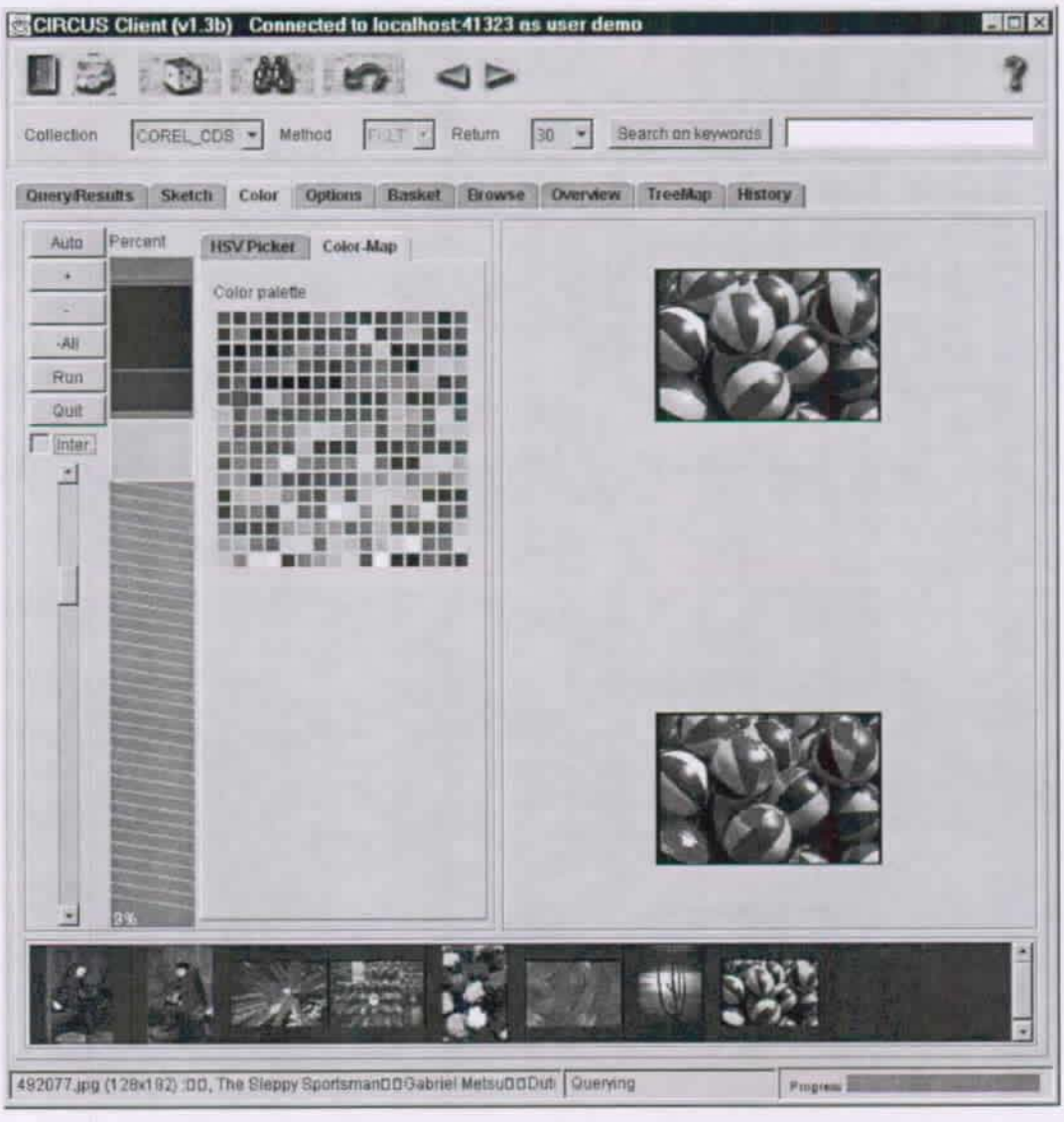


FIGURE 7.4: The query by color proportions. The colors are either selected from a system palette, entered by values or picked from a sample image (right-hand side of the screen).

Query by sketch A skilled user, like a graphics designer, could want to specify queries by constructing an example image by sketching the objects it should contain, and by pasting together pieces of images from diverse sources, in order to create a collage query. A query interface that allows for sketching or editing of images with simple tools can greatly enhance the expressive power of the system. The query formulation is identical to the Query by Example paradigm where the constructed image is either considered a positive or negative relevance element. Alternatively the positions of the pasted image snippets or sketches could be used in a spatial constraint scenario[1]. The editing functions are basic: freehand drawing, cut & paste from other images, flood-filling and smearing. In addition, the sketch tool can be used for assisted outlining of objects and annotation. Figure 7.5 presents the CIRCUS implementation of the sketch tool.

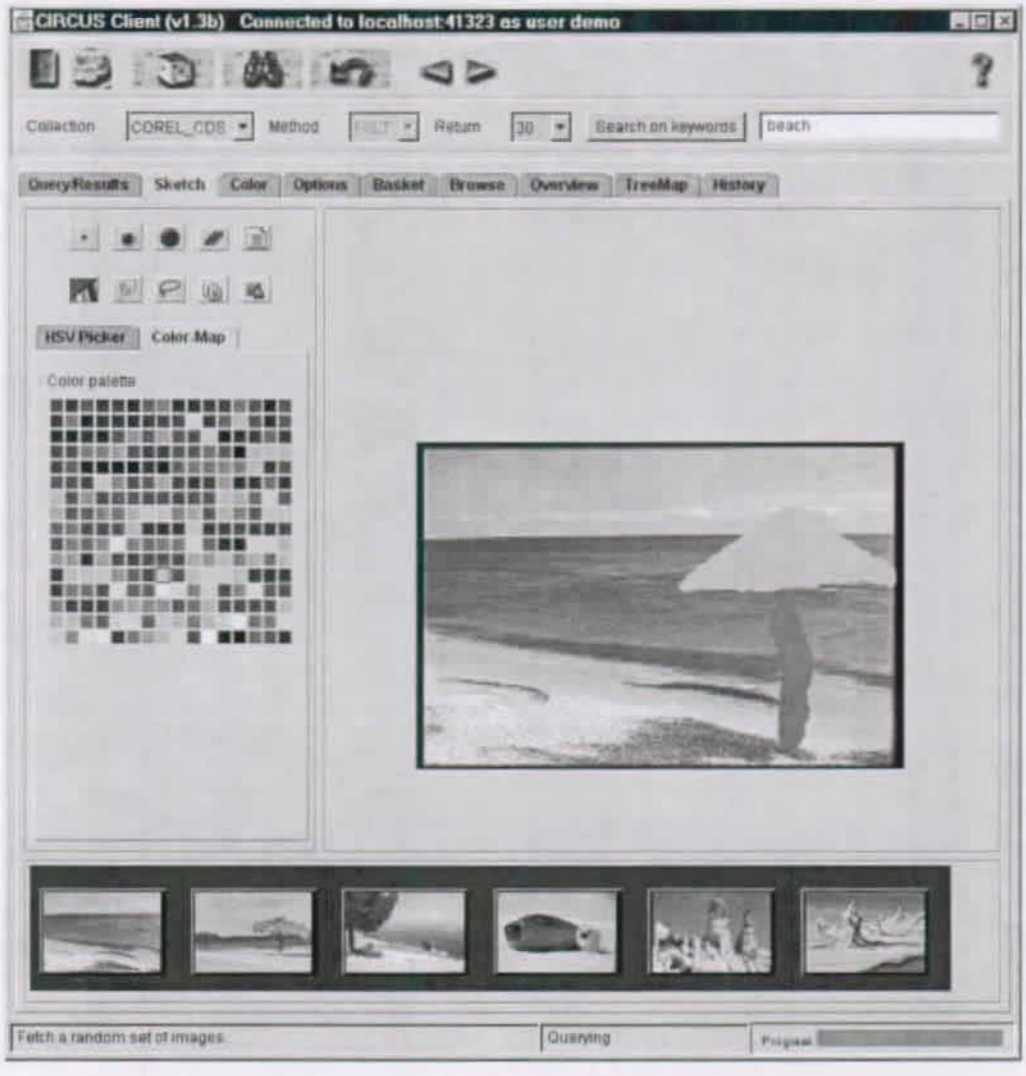


FIGURE 7.5: The query by sketch interface.

[1]The methods for retrieval we implemented do not explicitly provide spatial constraints, however the interface is capable of formulating such queries and sending them to the server using the MRML protocol.

Query by texture Certain applications require retrieval by texture properties. Examples include aerial photography, remote sensing, fabric catalogs, medical imagery and others. The user should be able to specify either generic properties of the region texture or samples of the texture from a collection of system stored or generated textures. Combining these aspects is an essential part of a functional and usable texture retrieval system. The mechanisms used to specify these properties are heavily dependent on the system's internal representation of texture. The matching function is likewise dependent on this representation. In CIRCUS we have several availible texture description mechanisms and the simplest one consists in comparing textures from a system catalog of textures to the patches of texture present in the images. Figure 7.6 shows a sample solution interface which allows for sample specification. The

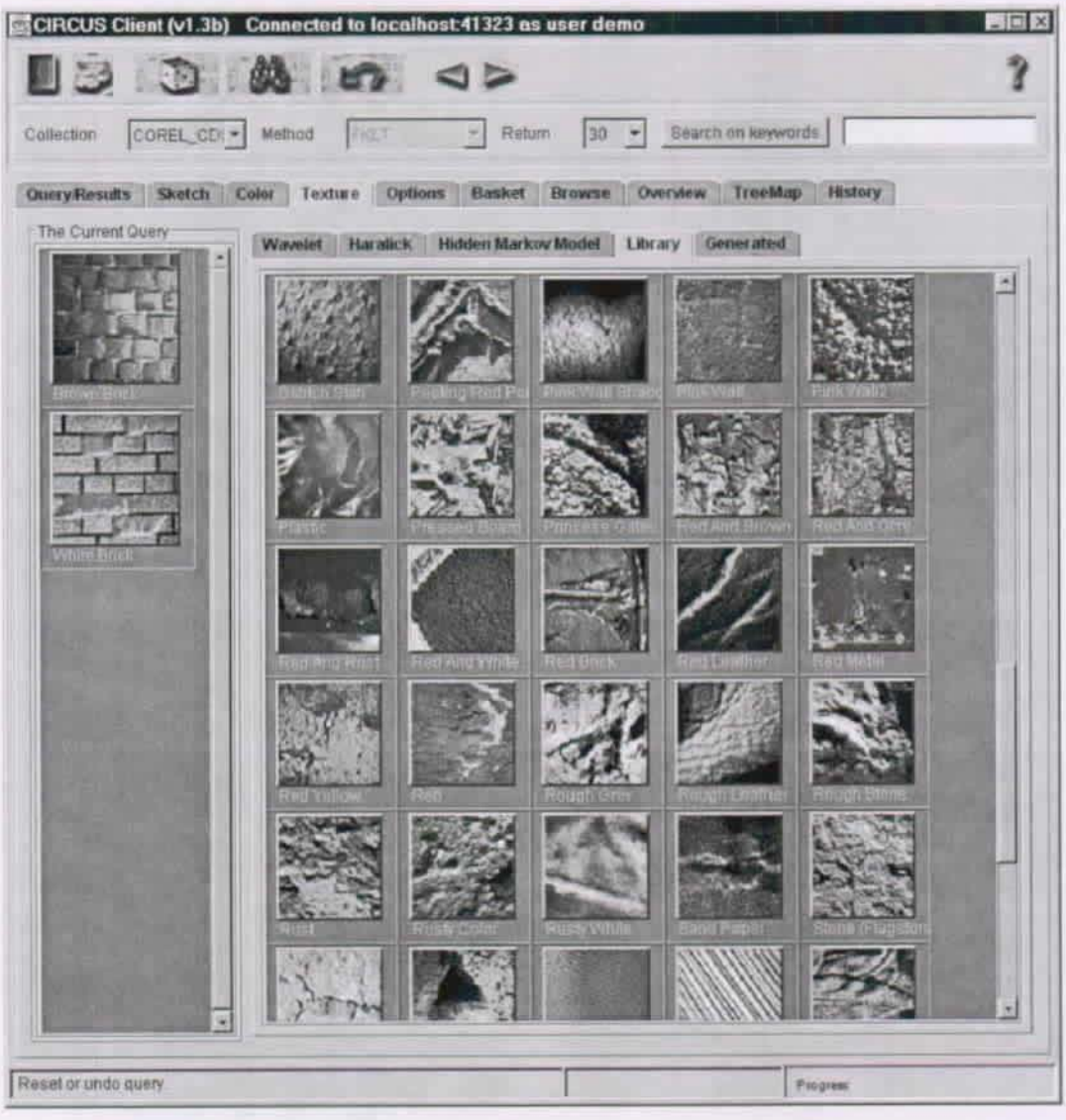


FIGURE 7.6: The query by texture interface. The selected samples on the left-hand side, the system texture catalog on the right-hand side.

other descriptive methods, when accompanied with a generation algorithm, can use the same interface. Some methods do not provide a generation of a texture based on the description, these methods are provided for by a specific interface which can only be effectively used by expert users that understand the description mechanism's parameters.

Query by annotation As mentioned before an important part of a multimedia database is the interaction between the textual annotation of an image and the visual aspects of the same. Any system that wants to support both textual and visual features must provide the user with flexible means of associating annotation to a query or image part. In many locations in the system, the user can select the option **Annotate** from the context menu (right mouse button) and enter an annotation for an image, a sketch, a color or any other part of the query. Formally

$$q = \{(q_1, t_1), \ldots, (q_n, t_n))\} \quad \text{where } t_i \text{ are the associated strings of text.} \tag{7.8}$$

The method applied by the system to exploit this information within the matching functions was explained in detail in Chapter 6. The interface then presents the annotated object with an overlaid icon (), as might be visible from Figure 7.7.

FIGURE 7.7: The query by annotation. Notice both the text-field in the upper-right corner of the interface and the overlaid icons on several images in the QbE/result Panel.

Additionally, a text-entry field is always available for the user to add textual concepts to the query. Likewise, a field in the query options allows the user to specify an URL that is used as a text document query element, in other words a very long text string. The image elements on the document are examined and treated as positive query examples, the text is treated as if it were the annotation of the images within the document.

Composite queries A final query paradigm is the combined query where any of the above mentioned query paradigms is used in conjunction with others. The query by annotation is as designed an integrated aspect of the previous paradigms. It can be used alongside with visual aspects or property queries. In order to facilitate the construction of combined queries the system should offer a simple interface, not based on complex algebras or query languages. It should remain a visual specification and be immediately clear. Formally the combined matching function is given by:

$$\text{match}_{\text{combined}}(I, q) = \text{match}_{\alpha}(I, q) \odot \cdots \odot \text{match}_{\omega}(I, q) \tag{7.9}$$

where match_{χ} is one of the previously defined matching functions and $\odot$ is one of the logical operators: AND; OR; and AND NOT.

An attempt of producing an interface that offers such complex tasks as combined queries, while respecting the KISS[2] principle is given on Figure 7.8. Any query element (sketch, color proportions, etc.) can be made active or inactive, it can be moved around on the display using simple graphical operators $\leftarrow, \rightarrow, \uparrow, \downarrow$. These structures the query into a tree. A query element with children (on the same horizontal line) will return results if and only if its children also return results. This implements a logical conjunction. A query element at the same level as another (on the same vertical line) will return results regardless of other queries at its same level. This implements a simple logical disjunction. A query element marked as negative (red border) will return all results except those corresponding to the element, in other words a simple logical negation. A query element marked as inactive will be ignored.

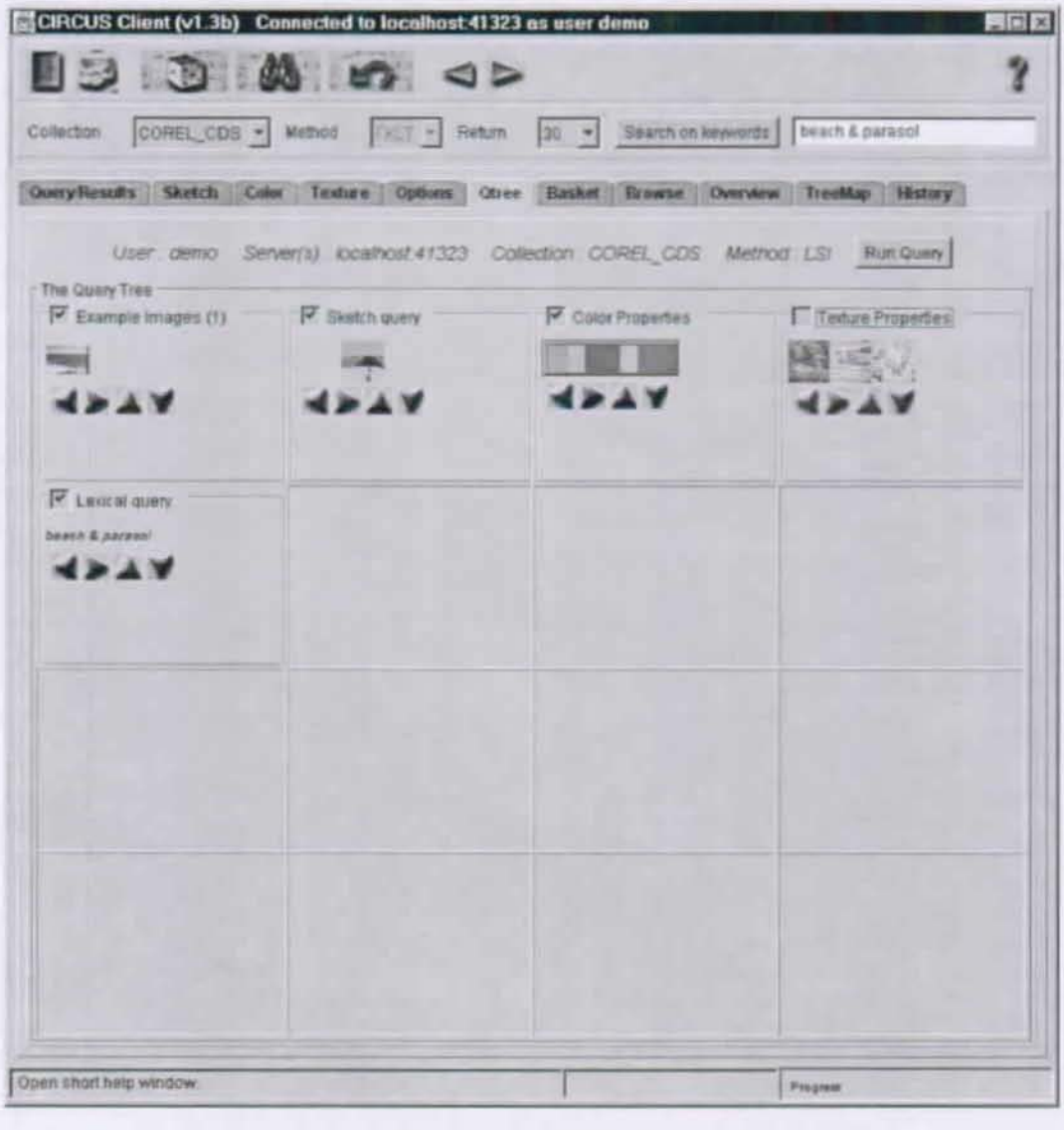


FIGURE 7.8: The combined query interface.

[2]Keep It Simple and Stupid is a slightly naughty way of abbreviating the basic idea that any interface should be as simple as possible, foolproof.

7.4 Result structuring and visualization

Another aspect allowing for greater bandwidth of data exchange between user and system, thus enhancing the efficiency of the interaction, is real-time interactive visualization. In a typical retrieval application, the results are presented to the user in a list-like fashion. The system produces for every query a ranked list of relevant material, the ranking being an approximation of similarity. This full ordering of results is often the most efficient solution, but the effectiveness is wholly dependent on the ability of the ranking function to capture meaningful similarity.

Before we continue lets recall that the Latent Semantic Indexing (described in Chapter 5 and Chapter 6) can return both documents and terms as relevant to any user query. In the figures, we present cases where only images where requested, except when indicated explicitly.

We subdivide the visualizations into three classes according to complexity. The first group corresponds to economic representations of lists of results in a two dimensional display space. A second group is a midway solution: the representations of result sets for which total ordering is not available. The final group presents the results in the original search space, attempting to preserve the distances and relative placements of the results.

7.4.1 Basic result visualization

Usually for screen-space economy the sequence of results is tiled in a reading-order two dimensional table rather than a flat list. This can both be productive, as can be seen in the interaction patterns of more experienced users and counter-productive where users are confused with the wrapping effect. The natural complete ordering of the initial results is lost. Figure 7.9 presents a standard display scheme for both flat and reading-order mapped result lists.

a) Linear Mapping

Q	1	2	...	23	24

b) Reading-order mapping

Q	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17
19	20	21	22	23	24

FIGURE 7.9: Flat and reading-order mappings of ranked result list. The results are numbered from 1 to 24, and the query is marked as "Q".

Alternative solutions consist in tiling the results into a grid with a concentric placement. The closest documents are placed close to the grid origin. If the grid is regular, with fixed cell widths and spacing, we have a dense and economical representation of the two dimensional relative placement of query and results. A simple one dimension to two dimension mapping that achieves a compact concentric gird is the *spiral*, or *reflected spiral*. In the close vicinity of the query, these mappings can significantly improve the relative placement conservation. In other words, the coherence of the displayed results is maintained, at least as far as their closeness to the query is concerned. Figure 7.10 illustrates these two spiral mapping concepts.

In the reading-order and flat mapping, moving towards far-off results is always according to a single direction, in the last two cases, spiral and reflected spiral mappings, this moving away from the query is in two simultaneous directions horizontally and vertically. This can lead to some confusion and is a slight drawback.

a) Spiral mapping

16	15	14	13	12
17	4	3	2	11
18	5	Q	1	10
19	6	7	8	9
20	21	22	23	24

b) Reflected spiral mapping

Q	1	8	9	24
3	2	7	10	23
4	5	6	11	22
15	14	13	12	21
16	17	18	19	20

FIGURE 7.10: Spiral mappings of ranked result list. The results are numbered from 1 to (24), and the query is marked as "Q"

The reading-order mapping has been qualified as most natural by users who are acquainted with image-based systems, standard and usual WWW page layout, and in general computer window based environments (icon file lists). It however confuses novice users since the image directly below the query image is usually less relevant than the images towards the end of the first row. These less experienced users claim the reflected spiral mapping to be more natural, especially as it gives the user a first order overview of the structure of the results.

7.4.2 Trade-off spaces

The placement of elements further away from the query, especially among each-other and not in comparison to the query, gets ineffective with any one-dimensional to two-dimensional mapping. The most natural way to overcome this problem is to return results according to separable similarity rankings. For instance, a similarity based purely on visual aspects and a similarity based on structure or semantics can be returned as separate coordinates in a trade-off space. The two separable, but often correlated, aspects can thus be plotted along orthogonal axis. This method has the disadvantage of leaving the space arbitrarily sparse. Either compaction or abstraction methods can be used to alleviate this problem. Changing the query parameters and observing the movement of the results in this slightly more semantic representation, lets the user explore the system notion of similarity. On Figure 7.11, the user has selected a similarity according to color on the horizontal axis and layout on the vertical one. Eventually, other coordinate spaces like radial coordinates could be used. The query example was the image of a fire extinguisher featured on the right of the screen. As the user moves the mouse over the points representing images, a tool-tip opens up to show the image located at those coordinates. A larger version of the image is also displayed above the query image.

A similar approach can be used with two successive steps of the user query refinement and feedback loop, or with two very different queries. Any of the previously returned results can be mapped along the horizontal axis and the results of the current step along the vertical one. Each result is thus assigned a location based on its similarity to two different query steps (or queries). This allows the user to understand which kind of query suits more the information need she/he is trying to satisfy. So if one query was a keyword based query and the other a query by example, the combined results can be anticipated, by plotting out the comparison on the trade-off visualization. A simpler scenario involving two distinct QbEs is depicted on Figure 7.12.

Even though the approaches mentioned last allow for more flexibility, as already highlighted, they entail a high empty space ratio on the screen. The gain in accuracy of relative placement with respect to the mapping case is significant, but not sufficient, some artifacts are still highly misleading.

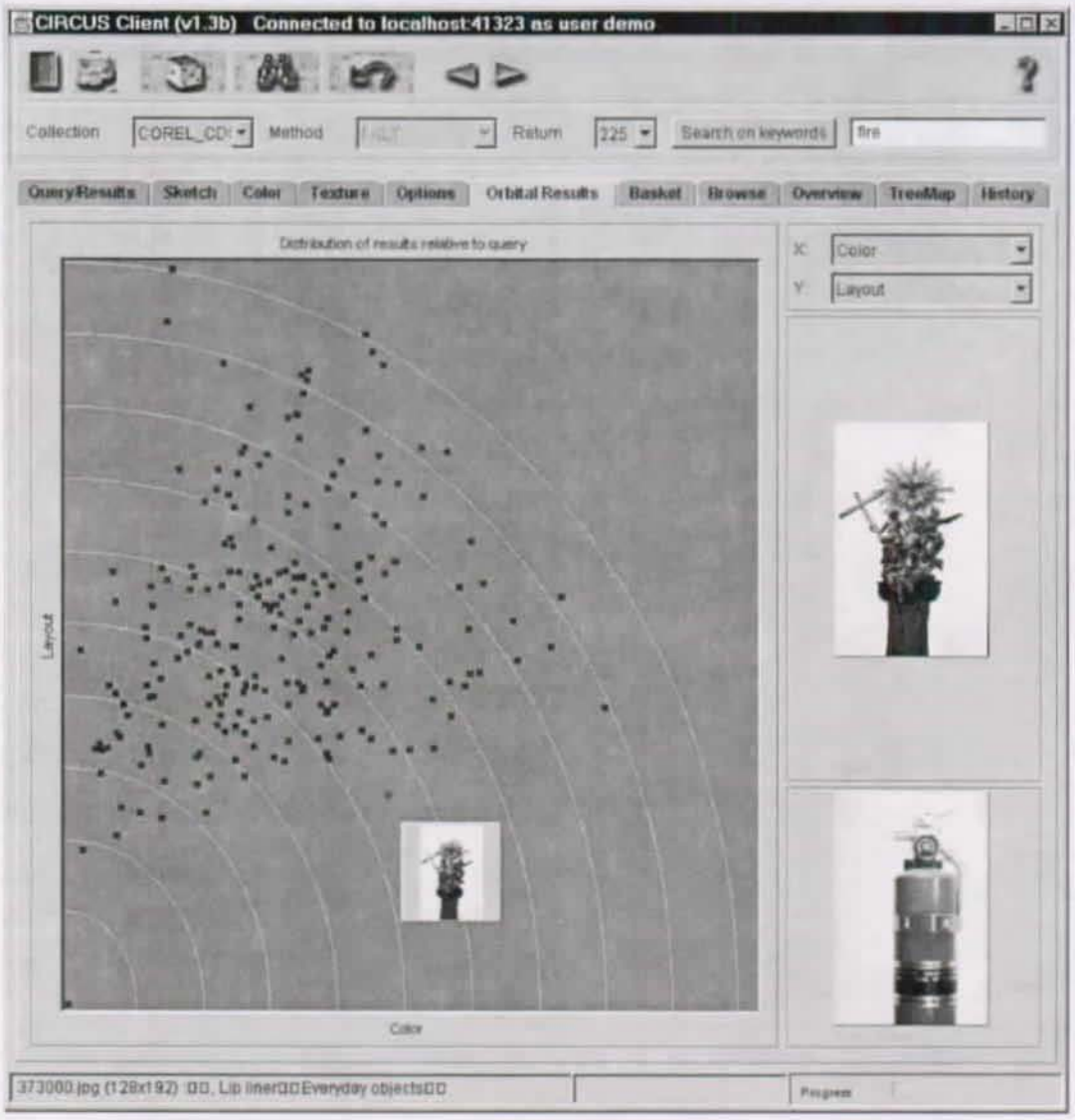


FIGURE 7.11: The results are presented to the user as a cloud of glyphs. The query is located in the lower-left corner. The position of a result along two user selectable axes is given by the similarity according to different aspects.

7.4.3 Distance preserving mappings

The final step to correct the relative placement problem is the immersion of the user into the similarity space itself. Since this is usually a high dimensional space (on the order of a few hundred dimensions), a direct representation is not possible. So a technique that preserves the relative distance information must be used to map a d-dimensional space to the screen coordinates. We have experimented with only two such methods. The first is the Sammon's projection J.W."nobreakspace –"Sammon (1969) which is detailed below. The second method is a standard self organizing map Kohonen *et al.* (2001), it is described on page 104.

Sammon's Projection

Given a set of N vectors $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ in $\mathbb{R}^d$, Sammon's projection (SP) attempts to find a set $Y = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$ in a lower dimensional projected space $\mathbb{R}^p$ (typically $p = 2$) such that the distances between pairs of vectors in X are preserved in their images in Y. Let us denote d^*_{ij} the Euclidean distances between $\mathbf{x}_i$, $\mathbf{x}_j$ in the original space and d_{ij} the distance between their images $\mathbf{y}_i$, $\mathbf{y}_j$ in the projected space. Sammon's algorithm tries to minimize the following error term:

$$E(Y) = \frac{1}{\sum_{i<j} d^*_{ij}} \sum_{i<j} \frac{(d^*_{ij} - d_{ij})^2}{d^*_{ij}} \tag{7.10}$$

FIGURE 7.12: The user is comparing results returned by two queries by example. The position of a result along the horizontal axis is the similarity to a previous query step, and along the vertical axis to the current one. The scaling has been adjusted to accommodate all results.

Minimization of $E(Y)$ is an unconstrained optimization problem of $N \cdot p$ variables y_i^j $(i = 1, \ldots, N;\ j = 1, \ldots, p)$. Such a problem typically has many local minima, thus a global minimum cannot be guaranteed using standard optimization techniques.

Sammon's algorithm uses a gradient descent iterative method to reconfigure the estimated Y so as to minimize $E(Y)$. The convergence rate and quality of gradient descent approaches depend heavily on the initial estimate of the solution, so an initial "guess" of Y is given by the principal component analysis projection to p dimensions. Furthermore, in order to diminish the influence of outliers, a normalization of X is used. Both variance and deviation from median have been used and the latter has proved more efficient. So we first switch to a normalized set of coordinates:

$$\mathbf{x}' = \frac{\mathbf{x} - m}{\psi},$$

where m is the median of the sample and

$$\psi = \frac{1}{N} \sum_{i=1}^{N} |\mathbf{x}_i - m|,$$

is the average absolute deviation from the median. Then the covariance matrix of X' is computed

$\mathbf{C} = X'X'^T$ and the p first eigenvalues $\mathbf{S}(l,l)$ $l = [1 \ldots p]$ and eigenvectors $\mathbf{U}$ are computed so that

$$\mathbf{C} \simeq \mathbf{USU}^{-1}.$$

Then we use

$$\mathbf{y} = \mathbf{Ux}'$$

as initial guess for the minimization critera in Equation (7.10).

An additional problem with the original SP is that it does not offer the generalization ability. When new samples are inserted into the collection, to determine their new positions in the map would require a re-run of the SP for the whole collection! For this, Mao and Jain (1995) proposed an interesting solution by modeling Sammon's projection using neural networks. In their model, the gradient descent optimization is converted into a back-propagation training.

We just briefly present the outcome of such a projection on Figure 7.13, more details on the interactions with the display can be found in Section 7.5.3.

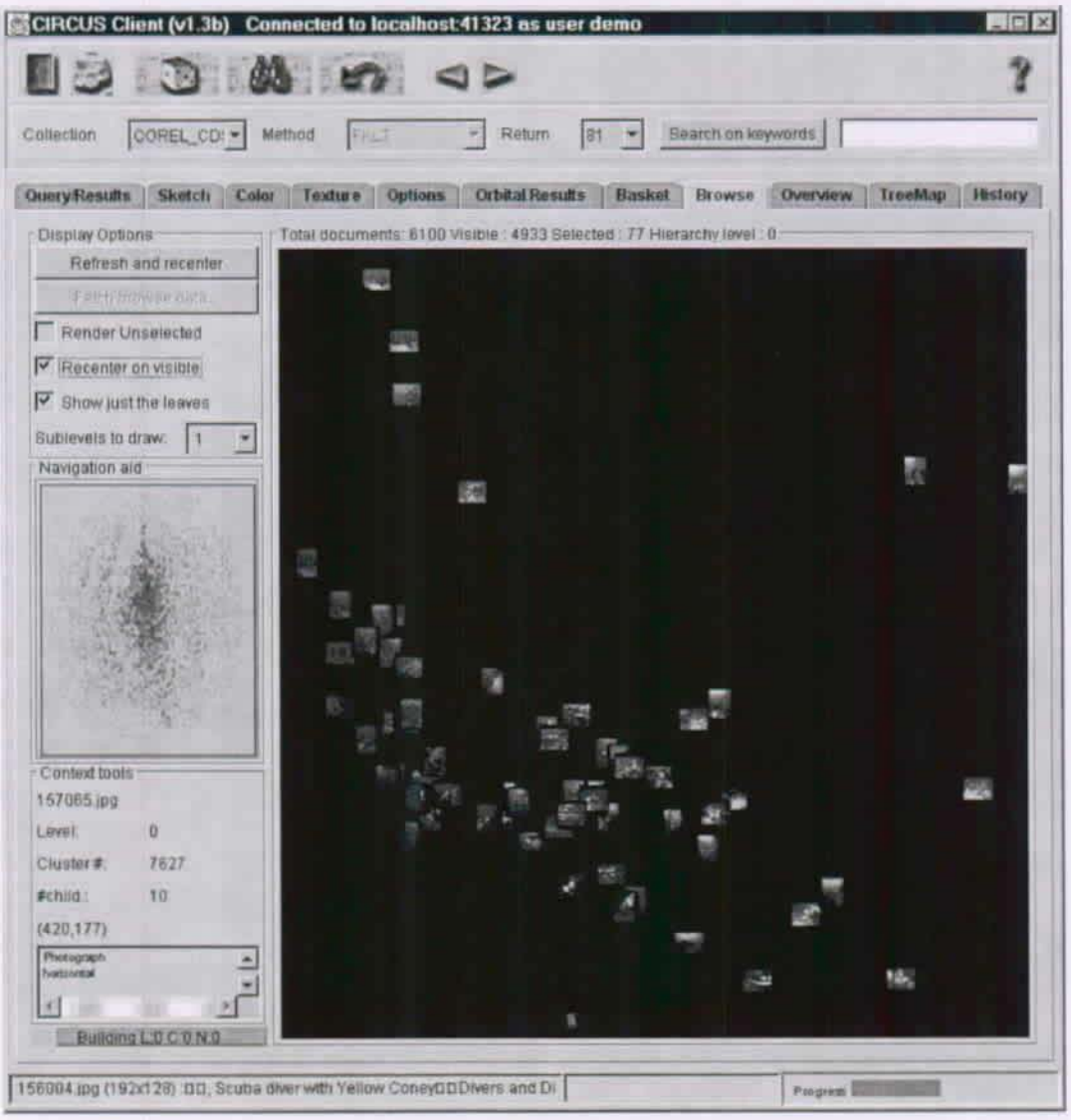


FIGURE 7.13: The query is the slightly larger image. The results are plotted at the estimated coordinates given by Sammon's projection.

Self Organizing Maps

Self organizing maps (SOM's), as described in Kohonen *et al.* (2001) offer much more flexibility than Sammon's projection. They can be seen as a form of vector quantization. They represent a large collection of documents with a smaller number of mapping units that have a fixed topological structure (neighbors and topographic distance are preserved under transformation). The advantages compared to Sammon's projection are: i) they provide for the projection of additional data, not taken into

account while computing the map; ii) they offer a more approximate yet more robust projection, iii) they contain an explicit topographic interpretation.

We keep the definitions for the N data points $\mathbf{x}$ as for Sammon's projection on page 102. The map is a set of K interconnected nodes or mapping units $\mathbf{y}$ that have basically three essential attributes: an input coordinate in $\mathbb{R}^d$, an output coordinate in $\mathbb{R}^p$ and a neighborhood connection $\text{Neighbors}(\mathbf{u}) = \{\mathbf{n}_i, \ldots, \mathbf{n}_f\}$. The number $K = m \times n$ of mapping units is an additional parameter of the algorithm.

The interconnection structure of the nodes is a planar graph, or surface, in the input and output spaces. The input space has dimensionality equal to the dimensionality of the problem. The output space has generally the dimensionality of the display metaphor (2 in our case). The mapping is initialized either as a uniform grid on the hyper plane defined by the first two eigenvectors of the covariance matrix of the data (PCA). It can also be initialized at random. Than a training algorithm performs a sequence of updates to the mapping in two successive steps. The coarse training moves nodes closer to the training data. Not only is the Best Map Unit (BMU) moved, but also its neighbors. The size of the influenced neighborhood is one of the parameters of the algorithm. In the fine-tuning step the size of this influence area is significantly reduced so basically only the BMU's are moved.

The final result is a set of interconnected nodes, that are still a manifold in both input and output spaces, but are no longer a necessarily smooth surface as initially (plane, torus, cylinder). The placement of the nodes tries to emulate at best the placement of the data points. Figure 7.14 shows such a map projected in three dimensions using principal component analysis. The map has stabilized after 5 minutes of CPU time for the rough training and an additional minute of CPU time for fine-tuning. The numbers correspond to timing on Intel Pentium II 450 Mhz processor running Windows NT and were programmed using the Matlab SOM package from Kohonen *et al.* (1996) and Vesanto *et al.* (2000).

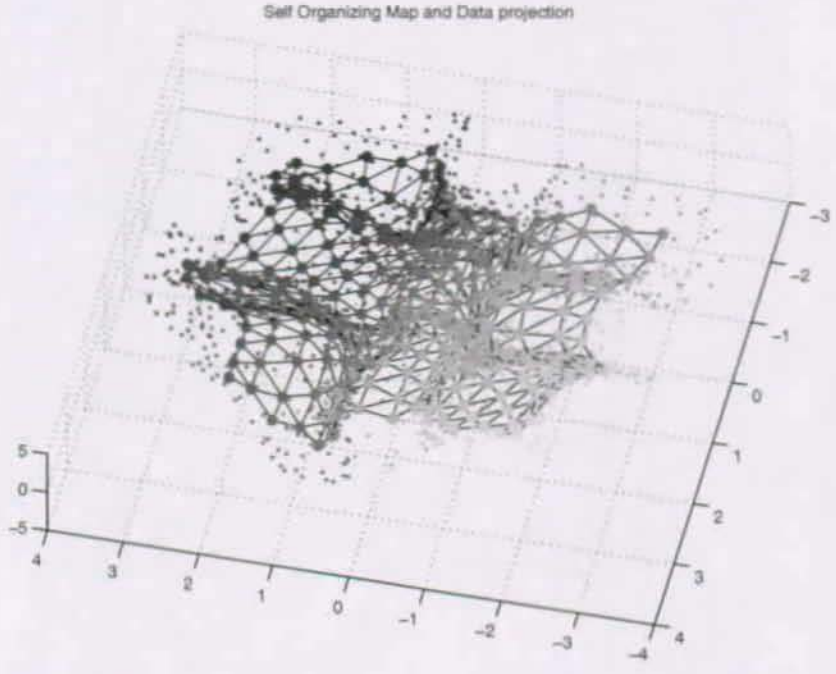


FIGURE 7.14: The SOM (spheres and grid) and data (points) visualized in the first three principal component dimensions.

Two measures of quality are often used to evaluate a trained SOM. The first is the average distortion:

$$E[D] = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{x}_i - \text{BMU}(\mathbf{x}_i, 1)\| \tag{7.11}$$

where $\text{BMU}(\mathbf{x}, k)$ is the k-th closest mapping unit in euclidean distance from $\mathbf{x}$. The second measure is the so called topographic error:

$$\text{T}_e(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} \Big(1 - \mathcal{N}(\text{BMU}(\mathbf{x}_i, 1), \text{BMU}(\mathbf{x}_i, 2))\Big) \tag{7.12}$$

where $\mathcal{N}(\mathbf{u}, \mathbf{v})$ is the neighborhood indicator function:

$$\mathcal{N}(\mathbf{u}, \mathbf{v}) = \begin{cases} 1 & \text{if } \mathbf{v} \in \text{Neighbors}(\mathbf{u}) \\ 0 & \text{if } \mathbf{v} \notin \text{Neighbors}(\mathbf{u}) \end{cases} \tag{7.13}$$

The topographic error is the proportion of data points whose first and second best mapping units are *not* neighbors on the map. In all our experiments the errors were $T_e \leq 9\%$ and $E[D] \in [0.6; 1.5]$. The latter being dependent on the data ranges itself but in general never exceeding twice the minimum distance between two *data* points. Typically the map will have significantly less nodes than the original set of data points (40×40 in our examples). In a second phase, a dimension reduction technique like Sammon's projection can be used to project the map into two dimensional display space. This is the approach we adopted.

Again, a simple outcome of a SOM followed by Sammon's projection is given on Figure 7.15, more details on the interactions with the display can be found in Section 7.5.3.

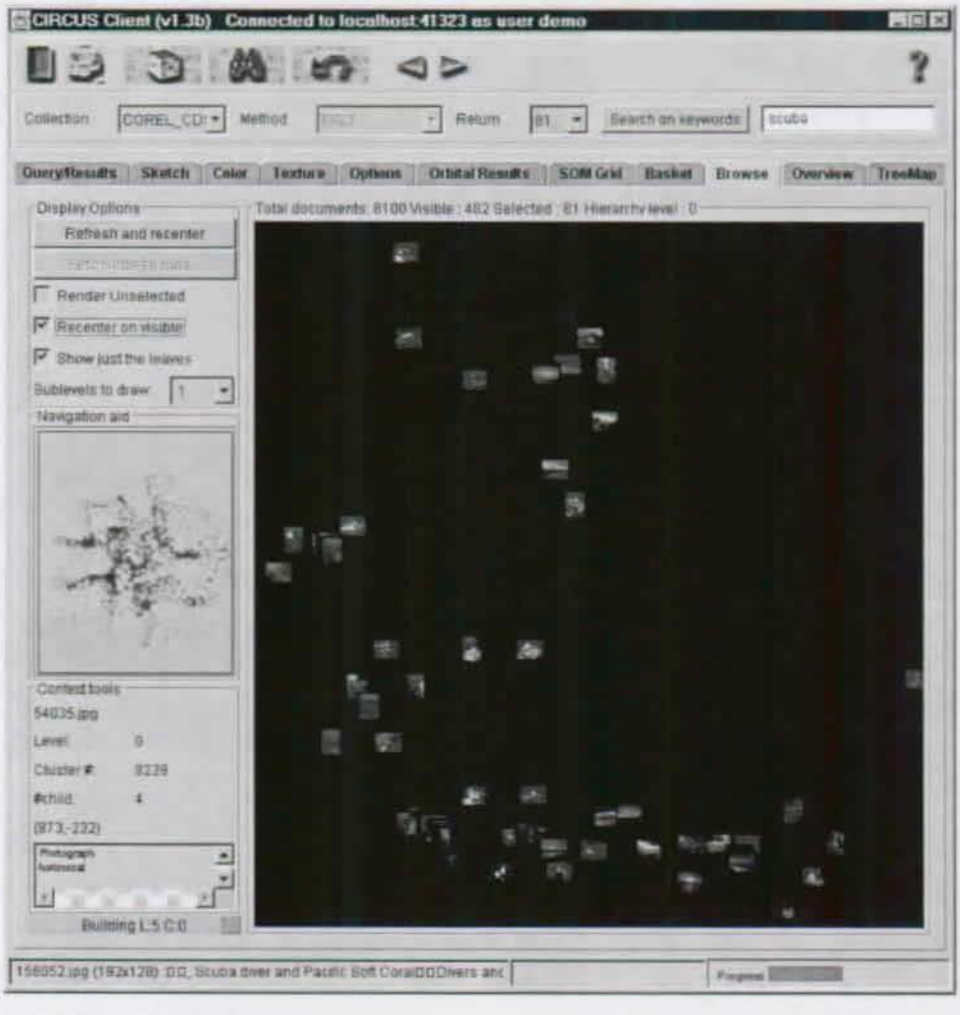


FIGURE 7.15: The same results as in Figure 7.13 are displayed using the two phase SOM and Sammon's projection mechanism.

Figure 7.16 presents the same results with both documents and textual terms returned. For the details of how this is achieved please see Section 6.4.

Figure 7.17 shows an analogous display where the SOM is represented by a compact square grid. Each cell corresponds to a mapping unit and contains a series of documents that are mapped to the cell. It also contains a series of textual terms returned by the LSI method. These are represented in the lower half of the right-hand side panel. After a query has been performed the highlighted cells are the ones that contain documents that where returned by the query. The level of highlighting is proportional to the number of relevant documents in each cell and/or to the activity of the cell. Clicking on a cell displays the contained documents in the right-hand side panel and the contained textual terms in the panel below it.

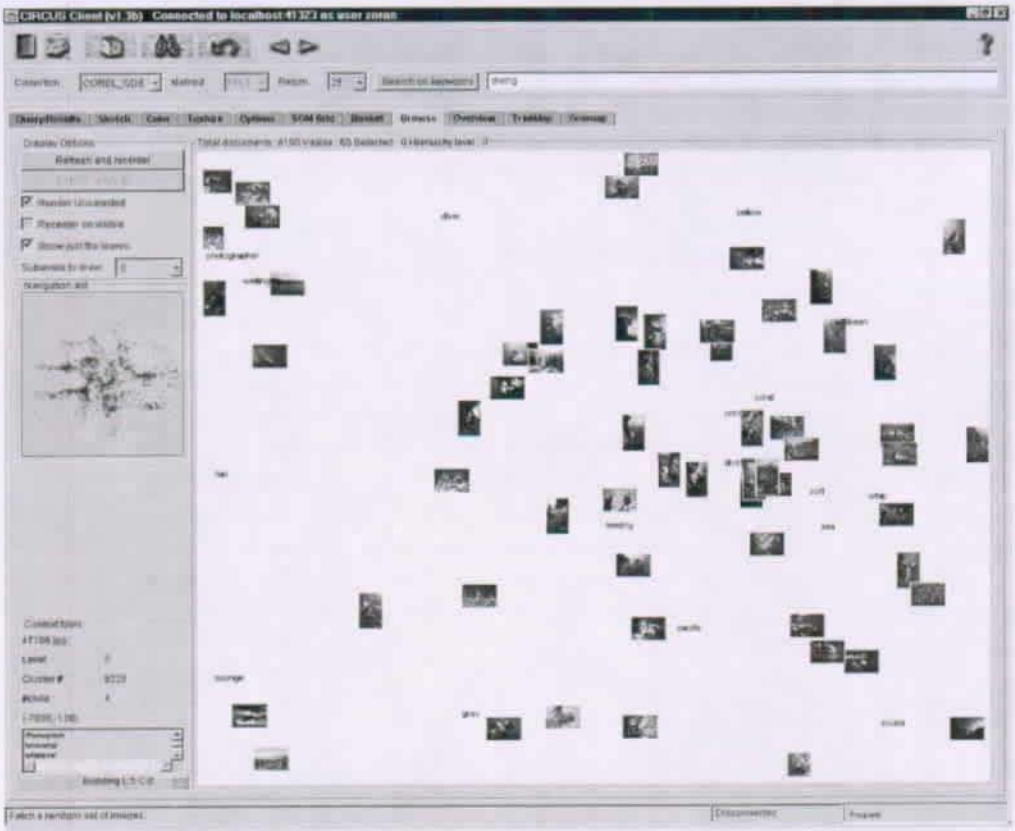


FIGURE 7.16: The same results as in Figure 7.13 but with textual terms returned as well as documents.

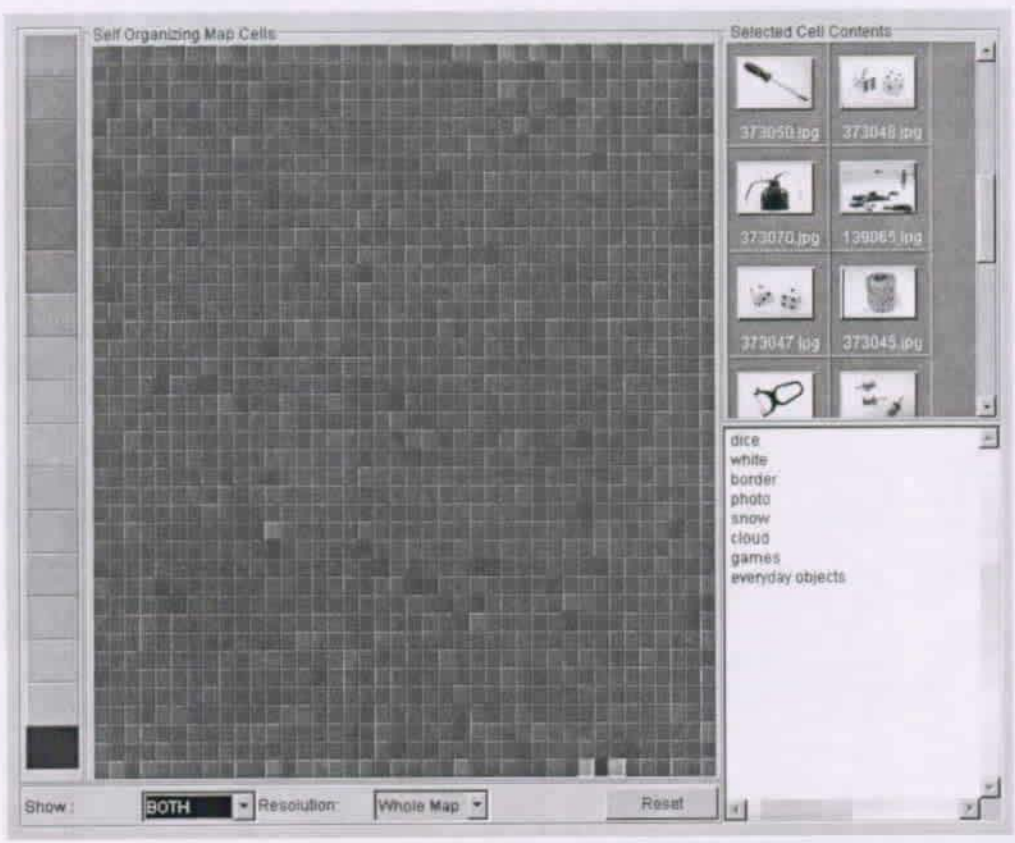


FIGURE 7.17: The SOM in intrinsic coordinates. Each cell is highlighted according to the proportion of relevant documents.

7.5 Conveying the collection structure

One source of user dissatisfaction with an information retrieval system is the lack of confidence in the system that novice users exhibit during interaction. Typical remarks we recorded while interviewing some novice users were:

> "How am I to know whether the collection doesn't contain other relevant documents than those returned?"
>
> "How am I supposed to interpret these results?"
>
> "Which are the criteria the system used to produce these results?"

We have investigated a set of techniques that can improve the effectiveness of the human computer interaction and lead the user to answers to the above questions. In the multimedia retrieval case, we developed interaction models targeted at getting the user familiar with the data collection and the system functionality. Interactive visualizations that convey the collection structure and the system view of similarity, matching and visual-textual integration, are the key to these concerns.

We present in Section 7.5.1 collection summaries based on semantic categories. In Section 7.5.2 similar summaries are generated but based on either visual or meta-data attributes. In Section 7.5.3, document similarity is used as the basic notion for the structuring into a compact representations. This sub-section also describes integrated modes for browsing and direct searching of the collection discussed in the task analysis of Section 7.2.

7.5.1 Category based overviews

Anyone using information technology agrees that we are overwhelmed by the quantity of information, yet are left starving for actual knowledge. There is a great need to make sense out of the huge amounts of available data, so much so that manual indexing, cross-referencing and classification are quickly becoming unfeasible. However automatic methods clearly have to respect our conceptions on meaning and integrate the widest possible range of semantic notions.

We have all been confronted with the dreary task of making sense out of complex data and interpreting complex configurations in the information we manage everyday. In a sense, we are used to tackling complex data and situations by decomposing them into simpler units wich are logically structured.

Multimedia retrieval is the prime application area of many modeling efforts. Going from a flat structured collection of documents, moving through a densely interconnected set of references to complete knowledge structures is unfortunately still a globally manual process.

One usually applied method is the structuring of the documents into categories. We distinguish several cases:

1. The categories are disjoint, meaning each document belongs to one and only one category. In this context, we also speak of classification.

2. The categories are overlapping, meaning that a document can simultaneously belong to several unrelated categories.

3. The categories are themselves structured.

Classification is useful in a few application domains, but in the general case either too restrictive or too application dependent. Overlapping unrelated categories can easily be transformed into a structured categorization. For this reason, we have concentrated only on this last scenario.

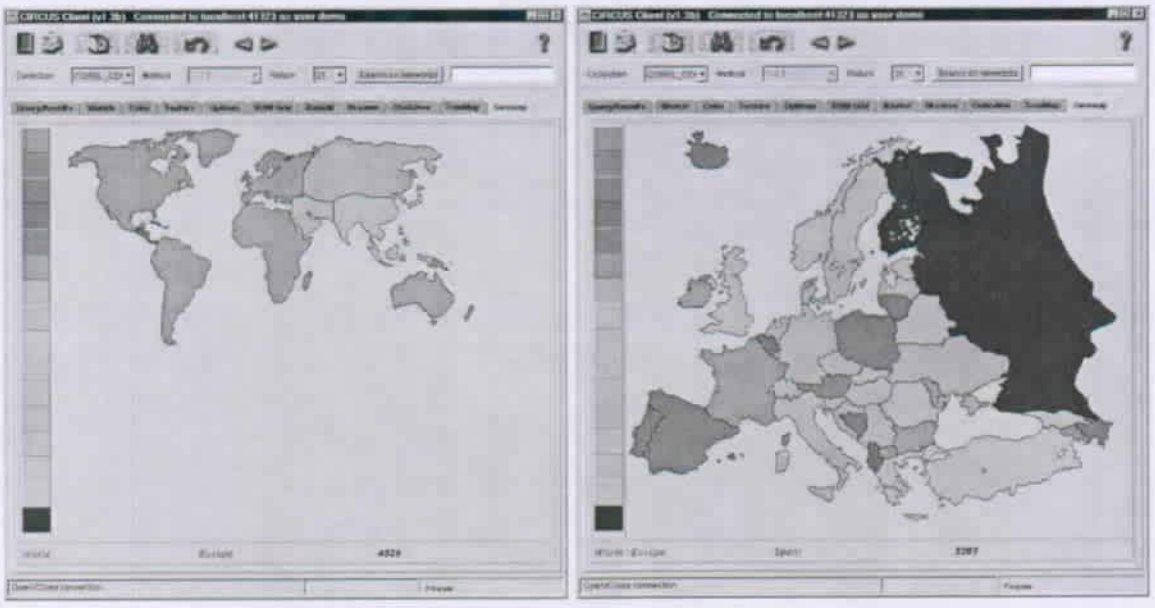
The documents we deal with are images, associated with annotations, so the categories are also a special type of annotation or associated semantics. Of course, according to the given collection the set of categories varies a lot. We illustrate our claims using the categories applied to the Corel Royalty Free image collection. More details on the data itself can be found in Section 6.5.

Among the many structures that could be imagined we illustrate and implement two types:

Geographic structure The images are all associated with a place of shooting or the provenance of the artist that painted, sculpted or build the depicted object. Each city is associated with a region, which in turn is associated with a country, itself part of a continent. A geography-based representation is given on Figure 7.18.

Thematic structure The images are annotated according to a specific subject matter they represent. These subjects are then grouped into more generic subjects, and these again into the most generic

subjects. The categories identified are a freely modified transposition of the open directory structure of the yahoo.com images directory[3]



(a) The entire world is represented. By moving over the continents the size of the sub-category is shown underneath the image.

(b) After clicking on Europe, the visualization changes to present only the map of Europe.

FIGURE 7.18: An overview of the collection structure in terms of geographic annotation. The color of a region is mapped to the proportion of images in the collection stemming from the region.

In the case of geographic structure a logical representation is a world-map with a specific coloring scheme that identifies for instance a property of a category of images, for instance the number of images it contains, the number of relevant images, or any other attribute of the images themselves that can be given a cumulative meaning. In other words, if the attribute of the parent node is the sum of the attributes of the children:

$$\text{attr}(\text{Parent}(d_i)) = \sum_i \text{attr}(d_i) \tag{7.14}$$

In both cases, the logical structure of the data is a tree. There exist various representations of large tree structures that always have to deal with a trade-off between complete representation,

visibility, and readability. Johnson and Shneiderman (1991) and Shneiderman (2002) propose the tree-map as an economic representation of a tree. The screen space (see Figure 7.19) is divided in alternating directions into slices proportional to the cumulative attribute with respect to the parent attribute. Then, at the lower level each slice is further subdivided in the opposite direction into its sub-node attributes.

If the space of each node is totally devoted to the sub-node representation we end-up with only the leaves or last branches represented. If on the other hand a frame is provided for each node before the subdivision then the entire tree is visualized. The basic algorithm suffers in the case of large differences in the tree structure and artifacts like thin long slices may appear. A squarified tree-map Bederson *et al.* (2001) solves this issue by heuristically determining the layout of each node along both directions. In this case a secondary visual cue must be given to guarantee that global structure is still visible. In Bruls *et al.* (2000) the authors suggest shading as a solution, but simply allocating additional space for a frame of the parent node alleviates this problem sufficiently. Figure 7.20 shows the same data as Figure 7.19 but using the squarified tree-map approach.

As in the geographic representation, the user can click on any region and it is expanded onto the entire display. The coloring of the regions can be used to represent a different attribute (not necessarily cumulative). In our examples the relevance to a given query is represented. Needless to say, the geographic interpretation can also be represented in a tree-map.

The user can explore the semantic organization of the documents in the collection and instantly have an idea whether the collection is likely to have relevant information to her/his needs. The compact

[3]See `http://gallery.yahoo.com` for more details.

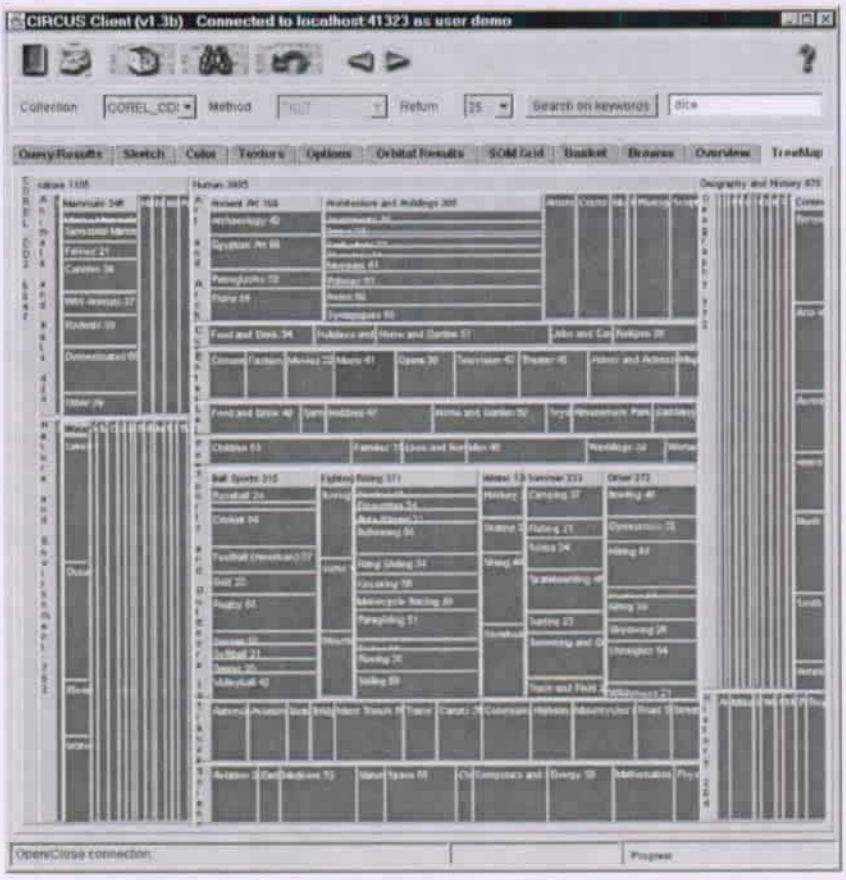


FIGURE 7.19: A classic tree-map implementation visualizing the number of children in each category. Notice the banding artifacts.

representations which convey a large quantity of information are likely to maximally exploit the highest bandwidth channel for communicating with the user, namely the visual channel.

A second type of visualization of the category tree structure can be achieved with a fixed tiling of the screen into regions. Each region contains a sample image from the category, the user can descend into the categories or request from the system alternative images in the same category. The choice of the images to represent are either random, based on medians of the image similarity in each category, or based on the closest images to the best mapping unit most representative of each category. Figure 7.21 shows this kind of representation.

In any of the above overviews, the system can map the color-codes, or other visual cues like size, to the relevance of the given category to a processed request. This way, the user can identify the categories most relevant to her visual query and decide to examine the entire contents of these.

Another important aspect not mentioned, but related to the overview and browsing modes of interaction is serendipitous discovery. While looking through an overview, the user might see a related, though not explicitly returned result to a previous query and see why this particular document was not returned. Section 7.5.3 deals in more detail with this idea of conveying the system's similarity judgment to the user and helping her/him understand its internals.

7.5.2 Attribute based overviews

A further aid, for grasping the structure of the database, is an overview of the documents, based on some characteristic. For instance, the user might like to know how many documents of a specific dominant color are in the collection. Or, for a range of categories, what are the dominant colors in the collection. Alternatively, the user might like to know what is the "distribution" of the image size with respect to some other attribute like number of contained objects.

We have developed three types of attribute overviews. The first shows the images as colored rectangles in a zoom-able and pan-able two dimensional space. The color is the average color of the image. The zoom-in to the colored rectangle eventually replaces the rectangle with the thumbnail of the image itself, once the number of documents to represent has reached a reasonable limit.

The positions according to the axes is given by numeric values of the selected attributes. When these

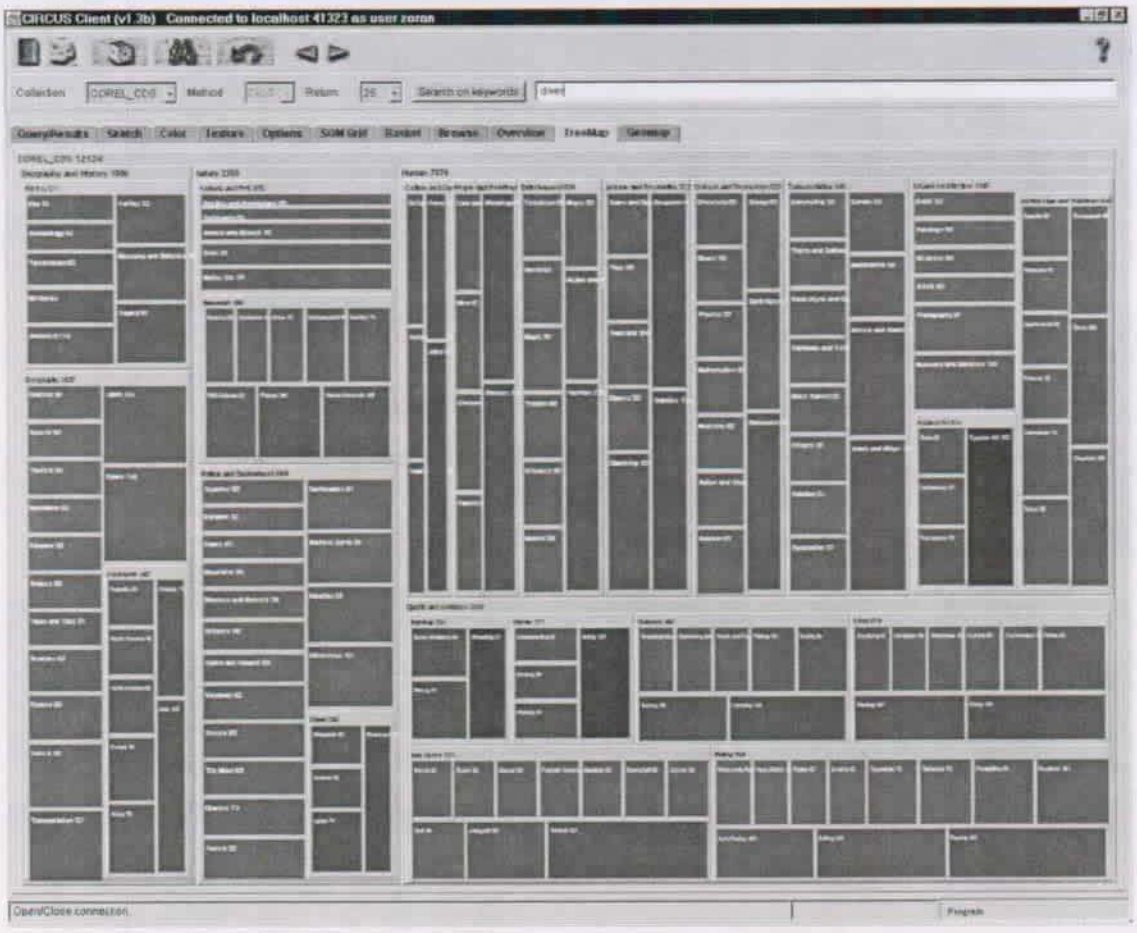


FIGURE 7.20: A squarified tree-map implementation of the same data as on Figure 7.19.

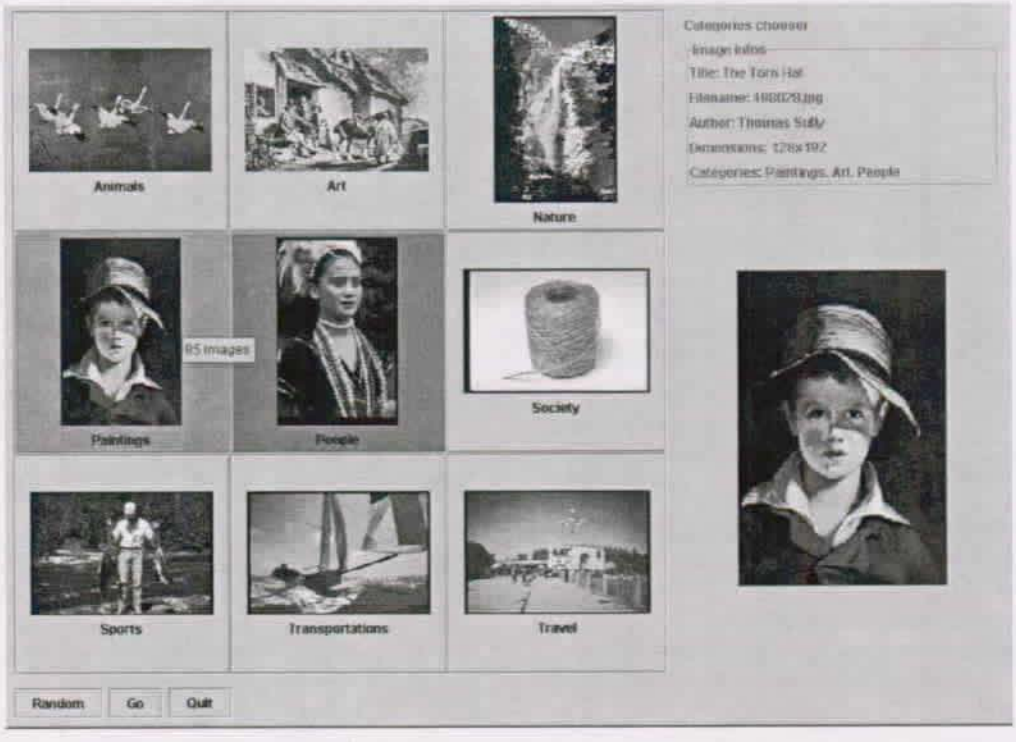


FIGURE 7.21: Each category is allocated a fixed amount of space on the screen and in it a representative image is shown. Clicking on an image changes the representative to a different one. Double-clicking on a category expands it to the entire screen and its sub categories are shown. At the last level a paged representation of all the contained images is shown.

are discrete, non numerical, attributes (categories or keywords), they are mapped to a set of bins and then randomly spaced out to cover three quarters of each bin range. The user is free to choose, from a large collection, which attribute to use on which axis. The implemented attributes are:

Color The average and dominant hue, saturation and brightness.

Texture The coarseness, directionality and contrast.

Complexity The number of segmented regions with respect to the average.

Annotation The image category and keywords.

Properties The image size, resolution, file size, dates, etc.

The structure of the database can thus be fairly quickly grasped, at least with respect to the chosen attributes. A third attribute can also be mapped to the colored rectangle size. Figure 7.22 shows the structure of the 6100 image COREL collection according to the hue and the first level category classification.

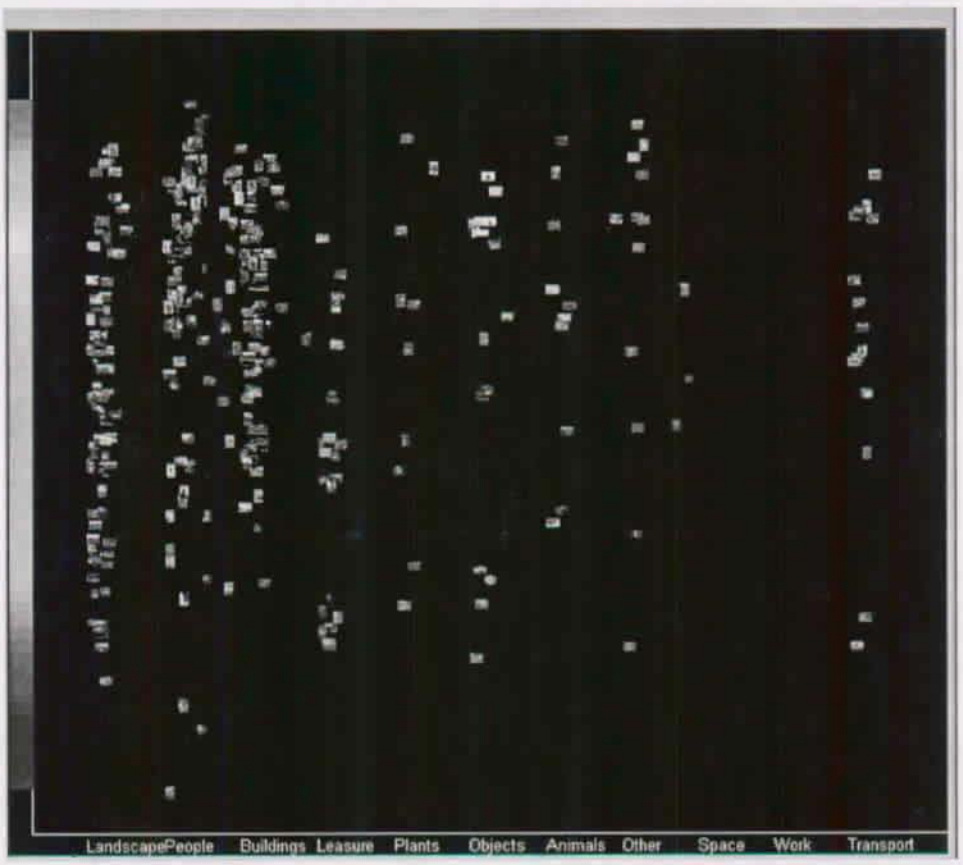


FIGURE 7.22: An attribute based overview. The user has selected dominant color hue on the vertical axis and principal category on the horizontal axis.

An overview based specifically on the color properties of the images was developed to represent the images in a layered fashion. The horizontal axis can be mapped to any of the three color coordinates in the chosen color-space. The vertical axis is attributed to another. Finally, the remaining coordinate can be either mapped by the size of

the glyph or can be used to create a layered representation. Namely, the mapping on the first two coordinates is carried out for all documents having the third coordinate in a certain bin. This is repeated for all the bins and the resulting representations stacked vertically or horizontally together. Such a representation is given on Figure 7.23. This approach can also be used with other attribute types, and lends itself well to discrete types like categories, or certain properties (creator, author, dates, etc.).

FIGURE 7.23: Color-space overview. The horizontal axis is mapped to average hue, the vertical axis to average luminance and the 3 layers correspond to three different saturation levels.

A more sequential overview must be used for larger collections, in this setting we cannot guarantee that all images would be easily shown in such representations as described above. A multi-resolution solution is given to this problem in Section 7.5.3, another simpler solution is to represent a single attribute and page through the bins sequentially, or to sample the collection and represent only a proportion of the images having that attribute value. A sequential solution is presented in Figure 7.24 where the images are sorted according to one of the attributes, and then represented as small squares of the dominant color. They are positioned using a reflected spiral, reading order, or spiral mapping (Section 7.4.1). The user can then page through the whole collection using the scroll bars. The center of the mapping can be switched to be one of the images and the rest can be sorted by attribute value difference from the selected image.

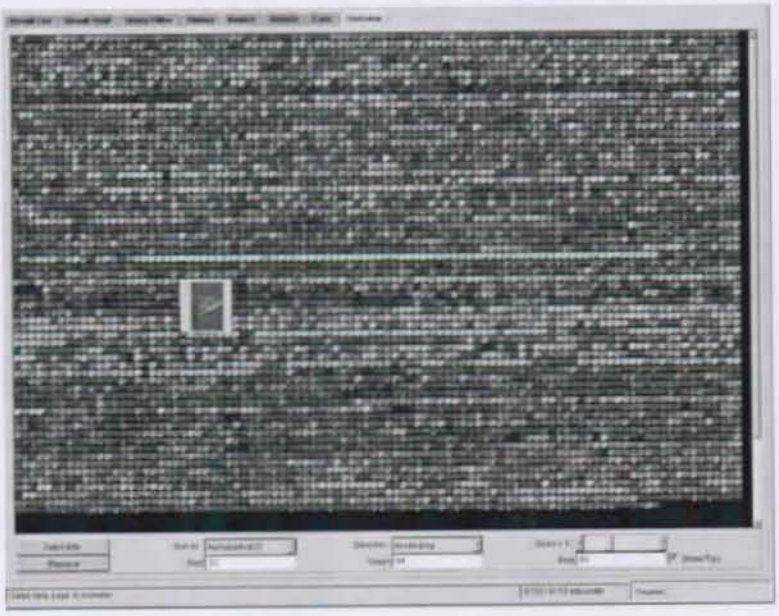


FIGURE 7.24: The images are sorted alphabetically according to their caption. The mapping is reading order and the number of glyphs per page is set so as to represent the images with the three most dominant colors.

7.5.3 Similarity based overviews

A further concern for many novice users is that of understanding why the system returns documents. Basically answering questions like:

> "What are the system criteria for judging relevance to my query?"
>
> "Why is this image deemed more similar to the query than this other image?"
>
> "How many images are as similar to the query as this particular image?"

Building on top of the result visualization ideas presented in Section 7.4.3 we present here similarity based overviews implemented in CIRCUS.

Starting from the notions of similarity or dissimilarity given for instance in Section 5.4.3, we would like to present the user with a schematic of the entire collection. We have seen that Sammon's method (Section 7.4.3) and SOMs (Section 7.4.3) allow a projection of the documents (and terms if LSI is used as underlying retrieval method). The placement of the nodes is such that the closeness on the display screen corresponds to closeness in the similarity space. The core idea is to present the entire collection to the user in this fashion. Unfortunately, projecting the entire collection is not a scalable solution for more than a few thousand documents.

We adopt a hierarchical clustering approach augmented with certain constraints like minimum and maximum cluster sizes. The display initially presents the top level cluster and possibly a few first levels of the tree. The user can select any node and descend into it to view more details. Furthermore, the display lets the user navigate in the projected similarity space. She/he can pan and zoom the display. The traversal of the tree structure — up and down the tree, and across sibling levels — is implemented by mouse clicks, and the spatial navigation with mouse dragging.

Figure 7.25 presents an interface enabling just that. The entire display is fully navigable: zoom, pan, tilt.

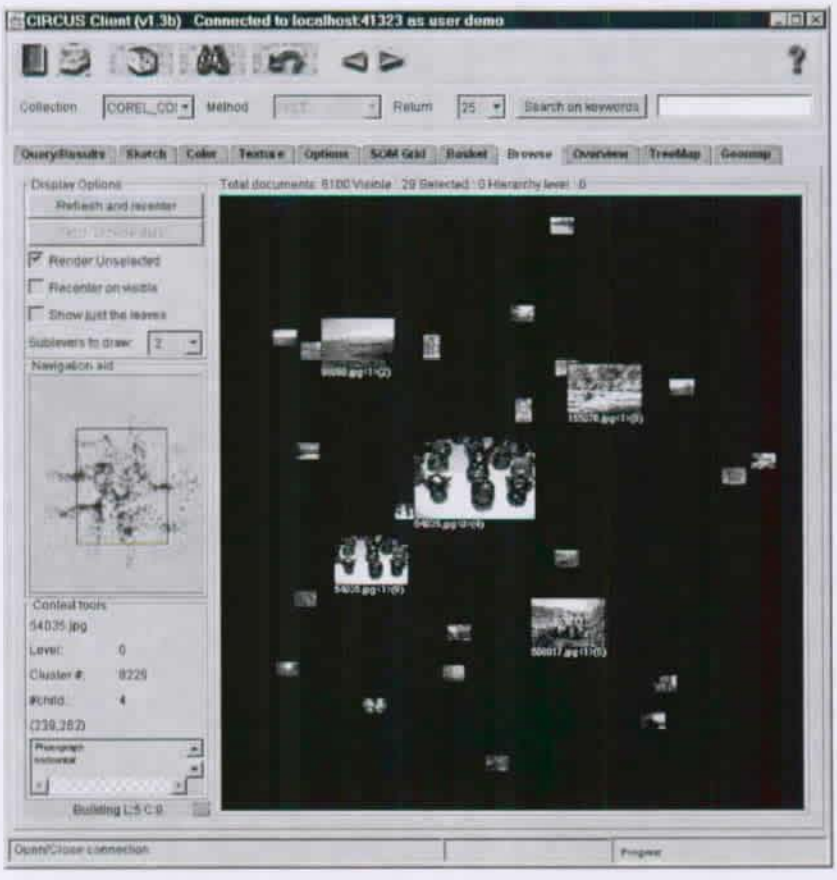


FIGURE 7.25: The first three levels of the hierarchy are represented through the decreasing size of the thumbnails. Notice the textual terms that are projected into the same search space as the result documents.

If the cluttering of the screen is not too high — less than a few hundred thumbnails — then the neighborhood of the examined region can be shown in all detail. Otherwise only a small number of sub-levels of the tree are shown.

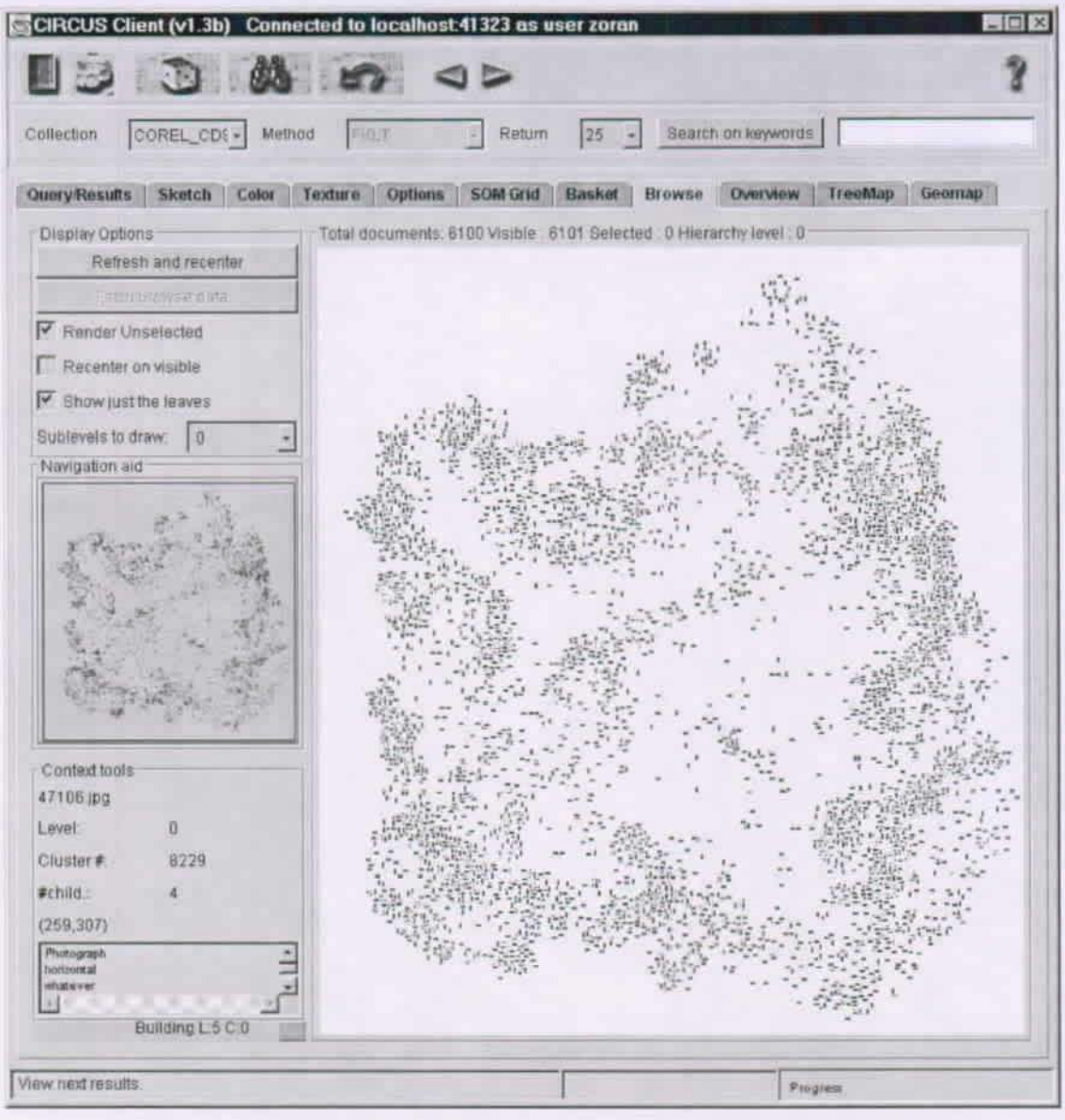


FIGURE 7.26: The entire COREL CD collection represented as a galaxy of colored points using the SOM method.

We provide for different "flavors" of similarity including color, gestalt, texture and annotation similarities, as well as any combination of these. The details of the associated dissimilarity functions can be found in Section 4.4.

A further option, the size of the collection permitting, is to present the whole collection as a galaxy of glyphs. Each document is thus represented by an abstract symbol. We chose to represent them as colored rectangles, the color corresponding to the average color of the image. This situation is depicted on Figure 7.26.

On Figure 7.27 we show a series of screenshots from a user navigation through the COREL collection using this interface. Notice also the projection of textual terms used by LSI on this figure and at the left of the screen the global map which shows the user which region of the search space she/he is exploring at any time.

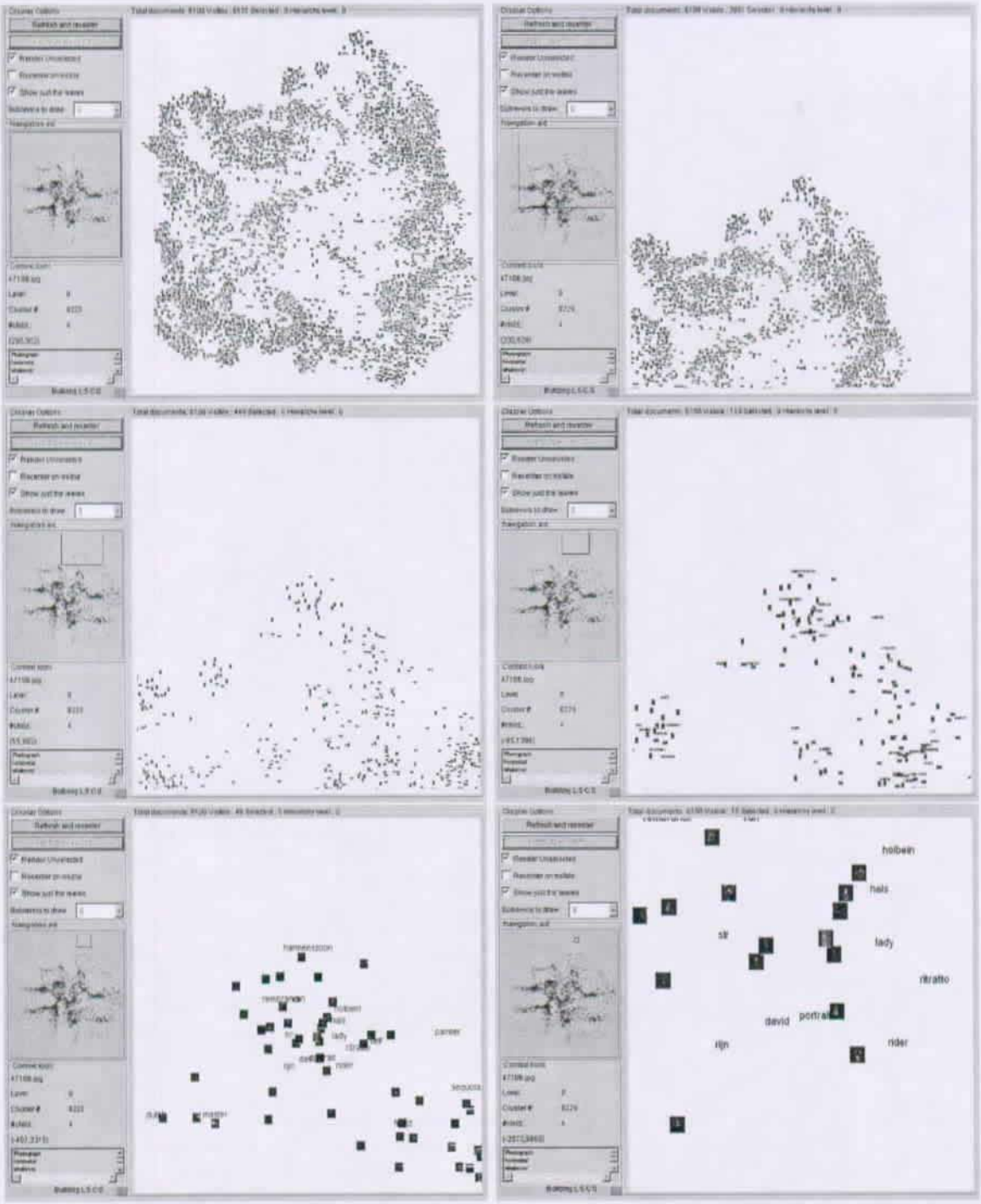


FIGURE 7.27: The user is navigating the overview by zooming in on a cluster of images. Notice the appearance of text terms as resolution increases.

Simultaneously, the display is coupled to a caching and multi-resolution framework which transfers detailed information from the data source to the client application as it becomes necessary. Annotations on the documents appear as the resolution increases sufficiently. Higher resolution images are fetched when interaction with the mouse has stopped and the expanded size of the thumbnails requires it.

In order to integrate a searching facility into this browsing mode, the user has access to a set of context dependent tools (panel at lower-left hand side of the display). When the focus in the search space is wide, the panel offers simple tools like keyword, color and image properties queries. When the focus becomes more restricted, queries by example become available. Finally when the focus is narrowed down to a very small number of images, image specific queries like sketches can be constructed. A scenario where the user has been offered the QbS type query, and has requested some similar images according to the modifications, is shown on Figure 7.28. The system has returned a set of images and displayed the first five in an overlay below the scatter plot.

FIGURE 7.28: An integrated searching facility in the browsing environment.

7.6 User - System collaboration

The collaboration of the user and the system is no doubt the best way to achieve the performance expected of the multimedia retrieval system. Clearly only the user can ultimately judge the relevance of the results. And however good and effective a system is it will never completely cover the range of human interpretation. In the case of image similarity, biased by the context in which the query is executed, the user can judge the similarity at a glance, especially detecting negative relevance. In the text retrieval case this is not so, since a reading of the document is necessary for the appreciation.

The system, on the other hand, is the only entity likely to have control and knowledge over the entire collection of documents. It can process large amounts of data with small delays, it can sort and reorganize the data relatively fast. So, a tight collaboration in form of an iterative process of query refinement and information need clarification is a crucial part of any effective system.

The two essential collaborative aspects we have concentrated on are described in the following paragraphs. To start out, we believe that effective satisfaction feedback[4] is of paramount importance for effective performance. Secondly, the querying interaction must be complemented with other interactive processes. We have illustrated the benefits of overview techniques. Similarly, seamless integration of searching and browsing brings high effectiveness improvements. The goal is to bridge the gap that separates the user's and the system's view on document similarity, result relevance, and completeness.

As already exposed in Chapter 6 the basic method we use offers a variety of ways to adapt the system judgment based on positive and negative samples. Ideally, the distance preserving methods of projection in Section 7.4.3 could be used to adapt the similarity judgments explicitly by the user. Vendrig *et al.* (2001) detail this procedure where the user is free to move the images around the display space into clusters, and lets the system come up with a distance measure that mimics this placement. It can be seen as an inverse Sammon problem, where the higher dimension space metric is adapted to reflect the lower dimension metric designed by the user. This new measure is then used to retrieve other documents. For the time being we use simple weighted distance measures that are updated heuristically.

We have been able to conduct only a few usability tests with expert as well as a few novice users. We got the general impression that often the feedback is meaningful in the first two iteration steps, and then no further substantial increase in performance can be achieved. Most users have agreed that actually seeing larger quantities of results, displayed in meaningful ways is more effective than classical relevance feedback on just a few results. The tradeoff visualizations (see Figure 7.12) that compare various query steps have notably been helpful in this regard.

The result visualizations respecting relative placement (see Figure 7.13 and Figure 7.15), can be viewed as semantic filtering of the global overviews. They seemed to motivate

users into exploring the neighborhood and finding results that would otherwise have been overlooked. The SOM grid view (see Figure 7.17) also gives the user many options for exploration around the returned results. Implicitly the user following these alternate interaction paths gains a lot of insight into the system's view point. Since the user can in a glance refute certain images as irrelevant to the query at hand, a low precision is not as bad as it would be in text retrieval or even video retrieval, where the document must be scanned at certain length to determine its relevance.

7.7 Summary and discussion

In this final chapter we presented some of our numerous experiments in the construction of novel interaction schemes for image+text retrieval. Some of these carry over transparently to audio or video retrieval, whereas others are media-specific.

The major contribution of the investigation is the integration of various query formulation methods and result structuring and visualization methods, into a smooth single immersive approach for browsing, searching, and navigating a large collection of documents.

The motivation was to increase the cooperation of the user and the system. On the one hand, the user should be able to comprehend the notions used by the system for relevance judgment. The system on the other hand, should adapt these notions as well as possible to the user. This cooperation is based on the principle that the human is the authority for deciding relevance, and that she/he can perform this much better, and even faster, than the system.

The still open issues, i.e. our wish list for future research, are:

- A more interactive and thoroughly tested satisfaction feedback loop should be implemented, perhaps following some of the ideas in Vendrig *et al.* (1999).
- A better interaction history and tracing tool should integrated to allow the user to manage "what-if" scenarios and branch into new directions without loosing previously achieved results.
- The lack of time and subject availability impacts on the quality of the scarce usability testing we have conducted. No doubt that with more input from potential users the system could be improved n many ways.
- The same reasons made an extensive qualitative performance measures impossible. The appreciation expressed by the test subjects was probably biased by their "technical" background.

[4]Relevance feedback is a term we avoid, not only due to its too frequent misuse, but also because it doesn't convey the complexity of the information that could be gained from monitoring user behavior and appreciation of the results.

7.A Details of the task analysis

This appendix presents the task analysis for the basic user tasks identified in Section 7.2. For each broad category: query specification, query execution, result visualization and browsing, we have given some scenarios that rely on the analysis of the tasks.

The hierarchical task analysis (HTA) as suggested in Annet and Duncan (1967) is used in a progressive and layered task description. The task tables that follow the diagrams describe the specific sub-tasks identified in the HTA with alternative execution sequences.

Query specifications

The user's basic task is that of finding a relevant set of documents, with respect to a certain information need. As we have already pointed out, this need can be very difficult to capture, and to express, especially for novice users. The criteria the user would employ to judge result relevance could be very abstract and very user specific. Thus the query specification offers the largest possible array of tools to meet the widest spectrum of user needs.

Scenario The user is looking for images that correspond to a specific set of criteria. The first step of specifying these criteria is to express them in terms of keywords, example images, visual characteristics and document properties (like size, age, resolution). The specification of the visual criteria like color or texture can be assisted by presenting the user with color palettes or texture swatches. The example images that the system can use for similarity retrieval could be either chosen from the current collection, using previous results or constructed by the more skilled users by painting or collage. All these specifications should then be combined using simple logical operations that the user should manipulate graphically. The user should not need to learn a query language!

On Figure 7.29 we present the HTA of the above scenario. The

convention of placing a solid thick line next to a task implies that the decomposition was not carried out in further detail. The dashed thick line with the underlying text indicates that the decomposition of that particular task is presented on another task graph. This was done for the sake of clarity, and later figures respect the same convention.

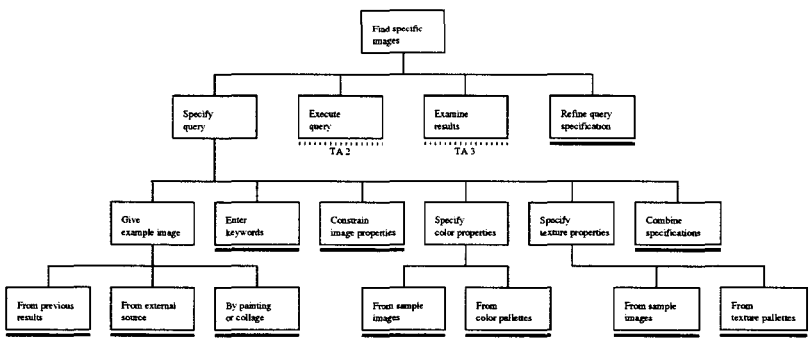


FIGURE 7.29: Hierarchical Task Analysis 1. The query specification.

In Table 7.1 we describe the identified tasks and sub-tasks, denoted with increasing indentation. The short description gives an idea of how the task si to be performed, and could be decomposed further into sub-tasks. These conventions are applied also to the following tables in this Appendix.

Table 7.1: The task table corresponding to query specification scenario. The sub-tasks are presented underneath the parent tasks with increasing indentation. They describe the alternative execution paths.

Task name	Task Description
Give example image	The user selects to represent the desired results using a set of sample images. She/he can specify positive examples and negative examples (i.e. images similar to the negative examples should *not* appear in the results).
from previous results	The sample images can be selected from a set of previously returned results visualized anywhere in the system.

Task name	Task Description
from external source	The sample images can come from a user specified source, like scanner, WWW, alternative document collection, local hard disk, digital camera, and so on.
by painting or collage	The sample images are constructed by the user using elements of images in the collection or images from externals sources. The painting and collage tools offer basic freehand drawings, and cut and paste operations. The paint color and brush texture should be selected from system pallettes.
Enter keywords	The keywords can be entered in an appropriate text-field, and will be used for semantic retrieval, but they could be also associated to any of the other visual criteria, like color names along color properties.
Constrain image properties	The basic image properties like size, resolution, dates, formats, etc. can be specified using numerical fields and selection lists or calendar tools for dates.
Specify color characteristics	The color characteristics of an image can be specified either in absolute or proportional terms. The first scenario uses presence or absence of specified colors, whereas the second scenario uses proportions of one color with respect to another.
from sample images	The colors could either come by picking the color of a region of an image already displayed to the user.
from color pallette	The colors are chosen from a system palette based on the most representative colors in the entire collection. The pallette can also be substituted by a color triplet specification in various color spaces.
Specify texture characteristics	A query can include patches of texture or patterns that can be provided using two methods: by example or by construction.
from sample images	A region of interesting texture can be selected, or an entire image of a textured patch can be specified from internal or external sources.
from texture palette	The texture properties can be specified by acting on parameters that generate texture patterns for preview, but that are directly used by the system to match for texture similarity.
Combine specifications	Logical combinations of the above specifications can be produced using a graphical metaphor. The query parts are layed out in a tree-like structure, where sibling elements are considered as logical disjunctions and child elements as conjunctions.

Query execution

This analysis illustrates the various ways a previously specified query can be executed by the user. Additionally implicit queries can be constructed by the Graphical User Interface in certain navigation and browsing tasks.

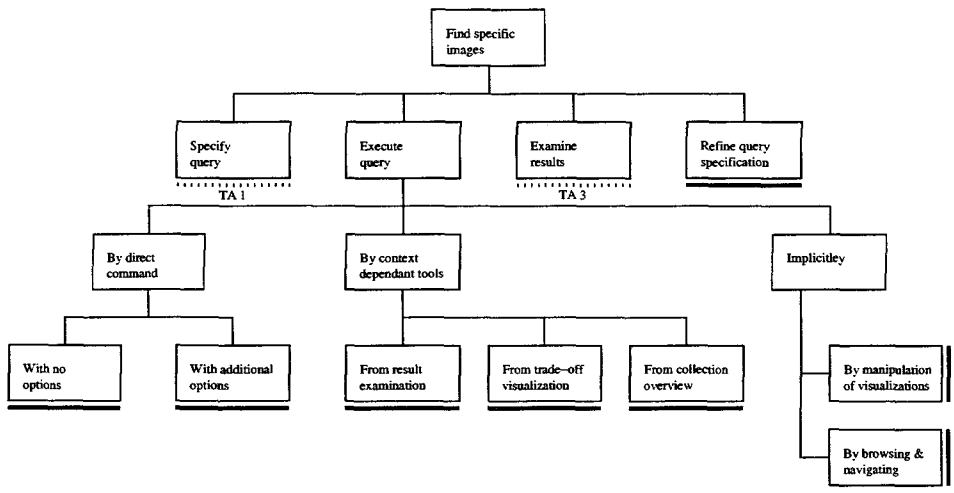


FIGURE 7.30: Hierarchical Task Analysis 2. The query execution.

Table 7.2: The task table corresponding to query execution. The subtasks are presented underneath the parent tasks with increasing indentation. They describe the alternative execution paths.

Task name	Task Description
By direct command	After query specification the execution of the query can be initiated using direct command.
With no additional options	The query is launched by double-clicking on sample images, or by selecting the appropriate menu item or execution button.
With additional options	The user first displays the query options tab in which additional query specific or algorithm specific options can be specified. Then the query is executed by pressing the query execution button, or by selecting the appropriate menu from the interface.
By context dependent tools	The execution can be launched explicitly from context dependent menus of the visualized objects.
From result examination	The user can select the appropriate menu item from the context menu (right-click) on each returned document icon or in detailed view from each document part.
From trade-off visualization	The user can re-launch previous queries from the various trade-off examination spaces by right-clicking on the displayed icons or axes, and selecting from the context-menu.
From collection overviews	Specific queries can be initiated using the context-menu from any element displayed in the collection overviews or browsing modes described below.
Implicitly	Implicit execution is launched when directly manipulating visualizations or while browsing.
By manipulation of visualizations	Moving objects and navigating the visualizations implicitly executes queries automatically constructed by the GUI and sent to the retrieval engine without explicit user intervention.
By browsing and navigating	Similarly while browsing the collection or navigating in collection overviews, the GUI implicitly constructs queries and executes them in response to user navigation commands.

Result visualizations

The following step after query specification and execution is the result visualization and examination. The following Figure 7.31 illustrates a possible hierarchical task analysis of these activities.

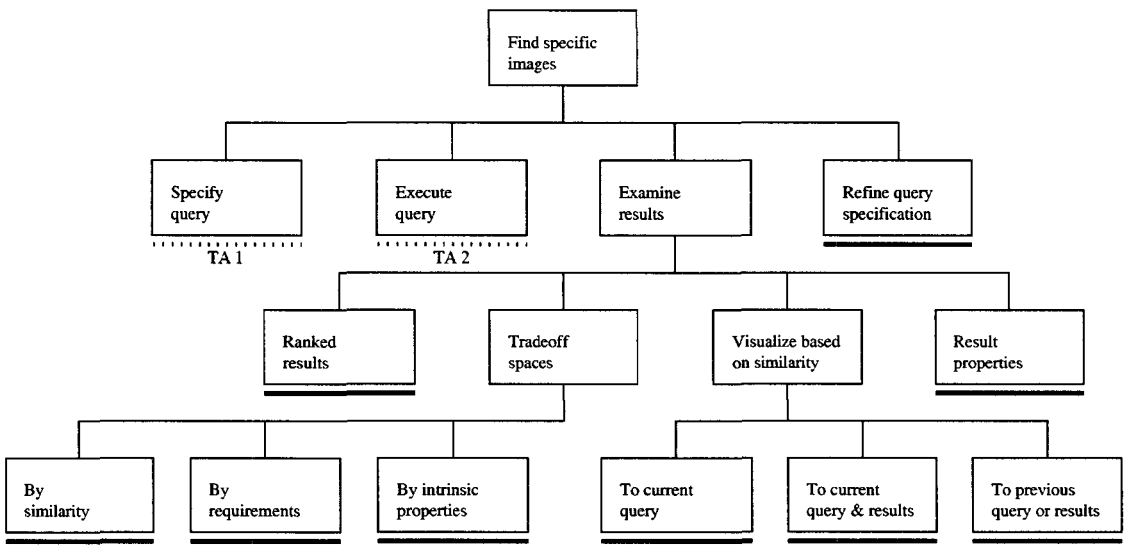


FIGURE 7.31: Hierarchical Task Analysis 3. The result visualization.

Table 7.3: The task table corresponding to result visualization scenarios. The sub-tasks are presented underneath the parent tasks with increasing indentation. They describe the alternative execution paths.

Task name	Task Description
Ranked results	The user examines the results as a ranked list. The order and mapping to screen space can be selected from pull-down menus.
Trade-off spaces	The user can examine the same results in different trade-off spaces. Two or three aspects can be compared in a scatter plot according two spatial and one highlighting dimension.
By similarity	The three independent aspects to trade-off can be similarities of the same set of documents to three different queries selected from the query history.
By requirements	The three "axes" can be mapped to similarities according to: different retrieval methods, different parameter settings or independent similarity aspects (color, texture, layout, annotation).
By intrinsic properties	The document properties (like size, format, dominant color, annotation, creation dates) can be used to map the results to any one of the tradeoff axes.
Visualize based on similarity	The ranked results can also be mapped in such a way as to match the spatial closeness to similarity.
To current query	The disposition of the results is such that the closer ones to the central query icon are the more similar.
To current query & results	Additionally results can be plotted at locations such that not only the closeness to the query corresponds to the similarity to the query, but also that the relative closeness is maintained among the different document icons.
To previous query or results	These visualizations allow the user to evaluate the impact of the change in parameter settings or other query specifications to the returned result lists. Here the difference in relative position and location of the previous visualizations is shown by trajectories.
Result properties	A final visualization is the detailed view of a single result document with its properties, relevant regions, annotation and statistics of similarity.

Overview techniques and browsing

The final analysis is dedicated to collection overviews and navigation. We present on Figure 7.32 and Table 7.4 the task decompositions and the task descriptions.

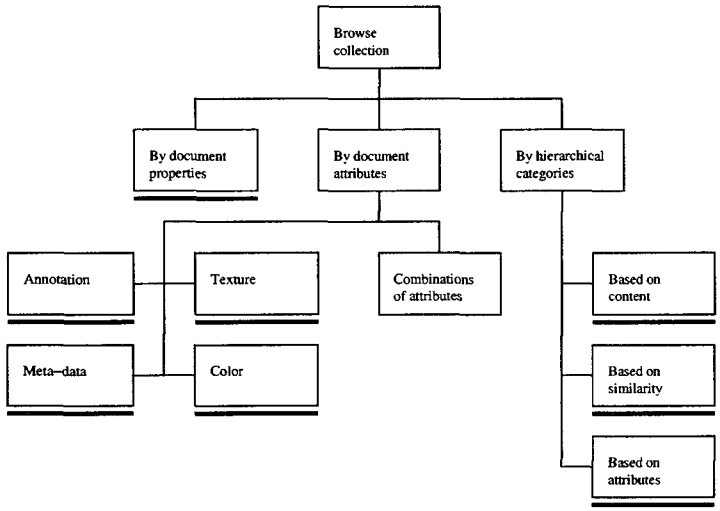


FIGURE 7.32: Hierarchical Task Analysis 4. Browsing the document collection.

Table 7.4: The task table corresponding to collection browsing and navigation. The sub-tasks are presented underneath the parent tasks with increasing indentation. They describe the alternative execution paths.

Task name	Task Description
By document properties	A summary of all the documents in a collection can be requested by the user, based on document properties. The display of the summary can be graphical (bar, pie or histogram charts), textual (lists or tables), or mixed (annotated graphs or illustrated tables).
By document attributes	The content of the documents can also be exploited by the user to establish synthetic views of a collection.
Annotation	The documents containing certain annotations can be extracted and visualized in a synthetic fashion (statistically, hierarchically or otherwise).
Meta-data	The extracted document content representations can be displayed as maps of the collection. This structuring is analogue to similarity browsing (see below).
Texture	The texture content of the image can be managed separately and a statistical or hierarchical overview of the documents containing similar texture characteristics can be clustered.
Color	Similarly to texture, color attributes of the documents can be clustered.
Combination of attributes	A statistical or hierarchical overview based on a combination of several of the above attributes can also be generated.
By hierarchical categories	If the collection has an inherent structure, or the document content can be grouped into a hierarchical representation the user can view the entire collection from this multiple resolution view-point.
Based on content	The depicted content can be grouped into categories, or the provenance of the images can be structured into geographical units and sub-units. Different visualizations of these structures can be displayed (tree-maps, geographical maps, tree-like lists, etc.) and navigated by simple mouse operations.
Based on similarity	The hierarchy can also be established and shown to the user based on similarity clustering. The user can navigate using simple mouse dragging and clicking through the representation both spatially and across different levels.
Based on attributes	The hierarchy for color and texture attributes can be used to create a navigable space showing representative images in locations of a color or texture pallette.

Chapter 8

Conclusions

The first chapter introduced the problems we wanted to solve, and the following chapters presented our investigations. Each chapter was closed by a discussion of the relevant results, but here we wish to highlight once again the principal contributions, the questions still open to research, and the implications of our findings for future research directions.

The three major axes along which our work has brought new insights and novel approaches to existing problems are i) the retrieval framework, ii) the retrieval method, and iii) user-interaction models.

8.1 Multimedia retrieval framework

The first contribution of our work lies in the system framework described in Chapter 2 and again in the communication protocol presented in Chapter 3. Much of the protocol investigation and design was done by Dr. Wolfgang Müller[1], Dr. David Squire[2] and the rest of Prof. Thierry Pun's Vision and Multimedia Lab of the University of Geneva.

Together we have proposed a communication protocol aimed at multimedia retrieval that could link together two retrieval components, a server and a client, over a network connection. The protocol was designed to be open, understandable and flexible. We have made available a set of predefined libraries for effortlessly managing the messaging, a retrieval method, a set of feature extraction libraries and a user interface tool.

We hope to involve more people on this project and make the protocol and message format a "standard" at least for the research community. The extensibility and inter-operability of components that exchange MRML messages lets the system designers and developers reuse tested components and examine new solutions to the same problems in a uniform framework.

Our own extensions to the current MRML specification reflect the system architecture we exposed in Chapter 2: a set of intercommunicating processing blocks. In this sense all components of the system exchange messages in MRML, and can be running on a cluster of different machines. The unification of these extensions and their integration into the MRML specifications is the major improvement left for future work.

The future of MRML is in the hands of its users, and the interest is growing daily. The rich discussions that are going on with researchers around the world has already brought many adjustments and extensions to the basic specifications, and we hope that this interest and interaction will continue to grow.

8.2 Image retrieval method

The method we proposed in Chapter 6, based on an established approach for text retrieval (Chapter 5), bridges the gap between visual and semantic characteristics of an image+text document. The interplay of these incommensurate and profoundly different aspects of information content is hidden in patterns of co-occurrences. These patterns are highlighted by a lower rank orthogonal transform of the matrix that represents the occurrences. Effective ways of exploiting these implicitly captured relationships can lead to new performance improvements, and especially to new functionalities.

[1]Now at University of Bayreuth, Germany

[2]Currently at Monash University, Australia.

The most important fact, once again, is that our approach uniformly integrates semantic and visual features into a single retrieval structure. The interchangeability of these features and the method's ability to retrieve not only documents but terms, allows for a mid-level semantics of visual aspects. Thus, a region of an image that has a certain combination of visual properties similar to a set of regions that have already been annotated as representing a semantic object, will implicitly inherit this association without user intervention. Conversely the method offers a visual description of semantic concepts, in the sense that a purely semantic query (e..g "red screwdriver") can be answered by documents or by terms. In this latter case the system can return only visual terms which best illustrate the semantic query (e.g. shapes, and colors).

We would have liked to test these last advantages of the method more, especially with rich data (see Section 6.5). The quality of the method relies heavily on the co-occurrence repetitions for similar terms in many documents, thus for a large collection of well annotated data it is more likely to have better performance than for a small collection.

Visual content characterization The retrieval method builds on top of classical feature extraction processing blocks, described in Chapter 4, which in our study were rather basic. We stress that our major goal was *not* to construct a series of top-performance image processing methods for visual content description, but rather to provide a proof of concept for their use in the retrieval method. However, we can mention the contribution of a series of improvements to an existing image segmentation method, and the uniform application of simple moment descriptions that allow effective shape and texture content capturing.

Obviously the improvements lie in the substitution of these basic content characterizations with more effective and more robust variants. Some pointers to such alternative candidate approaches have already been given in the discussions in Chapter 4.

Future research Additional efforts that could be devoted, and new tracks that could be followed for improving the retrieval method are:

- More adequate and effective visual content descriptions should be used.
- Vocabulary construction could be more robust with more effective clustering.
- More testing and a closer examination of the new functionality should be undertaken.

8.3 User interaction models

The last but not least set of contributions comes from our investigation of user interaction (Chapter 7). As any software system, multimedia retrieval is aimed at solving, or facilitating the solution of a specific set of tasks. These tasks can be executed using several scenarios. The basic one consists of a loop of information need formulation, result structuring and presentation, result examination and formulation refinement.

Our design of the user interface had in mind the first essential task of query formulation based on many different aspects: document properties, example documents, visual characteristics, semantics and combinations of the above.

The second phase, result structuring and presentation, was solved by providing meaningful and responsive result visualizations. The results can be structured using several relevance evaluations, and presented in several ways that reflect this relevance not only to the query but also among the results. The user can examine and manipulate these visualizations from many different viewpoints (structural and representational) and thus get acquainted with the abstract system notion of relevance, without having to infer it from complex parameter settings and algorithmic descriptions.

Linked to this aspect we find the notion of overview. A structured, synthetic and intuitive representation of the entire document collection, or for that matter of a subset like a set of results, increases user satisfaction with regard to the accuracy and completeness of the system response.

The refinement phase, incorporating result satisfaction feedback, has been given a minimal consideration. This is no doubt the foremost aspect to improve.

The fusion of query specification, result visualization, query refinement and overview techniques into a single search-space immersive approach, augments the system's learnability. It also offers the users an interaction metaphor, which is more directly physical and can draw from user intuition and automated behavior.

Let us just list the directions we would suggest to investigate more :

- More thorough investigation of satisfaction feedback (generally misnamed as relevance feedback). This would require also a further study of the retrieval method abilities for adaptation to the user's notions of relevance.
- Better exploitation of the interaction history for a single user session and for tuning based on multiple sessions/users.
- More detailed and diversified "what-if" scenarios where the user can balance and trade-off some properties of the results like visual and semantic similarity.

Bibliography

A. Pentland, R. P. and S. Sclaroff (1996). 'Photobook: Content-based manipulation of image databases'. *International Journal of Computer Vision* **18**(3), 233–254.

Ade, F. (1983). 'Characterisation of textures by "eigenfilters"'. *Signal Processing* **5**, 451–457.

Agnithotri, L. and Nevenka Dimitrova (1999). Text detection for video analysis. In 'IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL'99)'. Fort Collins, Colorado, USA. pp. 109–113.

Allan, J. (1996). Incremental relevance feedback for information filtering. In H.-P. Frei, D. Harman, P. Schäuble and R. Wilkinson (Eds.). 'Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'96)'. Zürich, Switzerland. pp. 270–278.

Alpkocak, A. and Esen Ozkarahan (1998). Similarity search in content based multimedia retrieval using spatial grid files. In C.-C. J. Kuo, S.-F. Chang and S. Panchanathan (Eds.). 'Multimedia Storage and Archiving Systems III (VV02)'. Vol. 3527 of *SPIE Proceedings*. Boston, Massachusetts, USA. pp. 155–166. (SPIE Symposium on Voice, Video and Data Communications).

Annet, J. and K.D. Duncan (1967). 'Task analysis and training design'. *Occupational Psychology* **41**, 211–221.

Antoine, J.-P. et al. (1997). 'Shape characterization with the wavelet transform'. *Signal Processing* **62**, 265–290.

Ardizzone, E., Marco La Cascia and Vito Di Gesú (1996). Content based indexing of image and video databases by global and shape features. In 'Proceedings of the 13th International Conference on Pattern Recognition (ICPR'96)'. IEEE. Vienna, Austria.

Ashley, J. et al. (1995). Automatic and semi-automatic methods for image annotation and retrieval in QBIC. In 'Storage and Retrieval for Image and Video Databases III'. Vol. 2420 of *SPIE*. pp. 24–35.

Aslandogan, Y. and C.T. Yu (1999). 'Techniques and systems for image and video retrieval'. *IEEE Transactions on Knowledge and Data Engineering* **11**(1), 56 – 63.

Asriel U. Levin, T. K. L. and John E. Moody (1994). 'Fast pruning using principal components'. *Advances in Neural Information Processing Systems* **6**, 35–42.

Atick, J. J., Paul A. Griffin and A. Norman Redlich (1996). 'The vocabulary of shape: principal shapes for probing perception and neural response'. *Neural Computation* **7**(1), 1–5.

Babu, G. P., Babu M. Mehtre and Mohan S. Kankanhalli (1995). 'Color indexing for efficient image retrieval'. *Multimedia Tools and Applications* **1**(4), 327–348.

Baek, D. H., Heui Seok Lim and Hae Chang Rim (2000). Latent semantic indexing model for boolean query formulation.. In N. Belkin, P. Ingwersen and M.-K. Leong (Eds.). 'Proceedings of the 23rd Annual International Conference on Research and Development in Information Retrieval (SIGIR-00)'. Vol. special issue, v. 34 of *ACM SIGIR*. ACM Press. N.Y.. pp. 310–312.

Baeza-Yates, R. and Berthier Ribeiro-Neto (1999). *Modern Information Retrieval*. paperback edn. Addison-Wesley.

Bartell, B. T., Garrison W. Cottrell and Richard K. Belew (1992). Latent semantic indexing is an optimal special case of multidimensional scaling. In N. Belkin, P. Ingwersen and A. M. Pejtersen (Eds.). 'Proceedings of the 15th Annual International Conference on Reasearch and Development in Information Retrieval'. SIGIR Forum. ACM Press. New York, NY, USA. pp. 161–167.

Bederson, B., B. Shneiderman and M. Wattenberg (2001). 'Ordered and quantum treemaps: Making effective use of 2d space to display hierarchies'. *ACM Transactions on Computer Graphics.*

Bellegarda, J. R. (1997). A latent semantic analysis framework for large-span language modeling. In 'Proc. Eurospeech '97'. Rhodes, Greece. pp. 1451–1454.

Bellegarda, J. R., J. W. Butzberger, Y.-L. Chow, N. B. Coccaro and D. Naik (1996). A novel word clustering algorithm based on latent semantic analysis. In 'Proc. ICASSP '96'. Atlanta, GA. pp. 172–175.

Belongie, S., Chad Carson, Hayit Greenspan and Jitendra Malik (1998). Color- and texture-based image segmentation using EM and its application to content-based image retrieval. In 'Proceedings of the International Conference on Computer Vision (ICCV'98)'. Bombay, India.

Beretta, G. and Stéphane Marchand-Maillet (2001). 'Benchatlon web site'. http://www.benchathlon.net/.

Berman, A. P. and Linda G. Shapiro (1999). Efficient content-based retrieval: Experimental results. In 'IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL'99)'. Fort Collins, Colorado, USA. pp. 55–61.

Berry, M. (1992). 'Large scale singular value computations'. *International Journal of Supercomputer Applications* **6**(1), 13–49.

Berry, M. et al. (1996). *SVDPACKC Version 1.0 Users Guide.* revised edition edn. University of Tennessee, Department of Computer Science.

Berry, M. W., S. T. Dumais %A and A. T. Shippy (1995*a*). A case study of latent semantic indexing. Technical Report UT-CS-95-271. Department of Computer Science, University of Tennessee.

Berry, M. W., S. T. Dumais and G. W. O'Brien (1995*b*). 'Using linear algebra for intelligent information retrieval'. *SIAM Review* **37**(4), 573–595.

Berry, M. W., Susan T. Dumais and Todd A. Letsche (1995*c*). Computational methods for intelligent information access. In 'Proceedings of Supercomputing'95'. ACM/IEEE. San Diego, CA.

Beylkin, G., R. Coifman and V. Rokhlin (1992). Fast wavelet transforms and fast algorithms. In Y. Meyer (Ed.). 'Wavelets and Applications'. Masson. Paris. pp. 354–367.

Bimbo, A. D. and P. Pala (1996). Shape indexing by multi-scale representation. In A. W. M. Smeulders and R. Jain (Eds.). 'Image Databases and Multi-Media Search'. Intelligent Sensory Information Systems, Faculty of Mathematics, Computer Science, Physics and Astronomy. Amsterdam University Press. Kruislaan 403, 1098 SJ Amsterdam, The Netherlands. pp. 43–50.

Boker, S. M. (1995). The representation of color metrics and mappings in perceptual color space. Technical report. Department of Psychology, The University of Virginia. Charlottesville, Virginia 22903.

Bouet, M. and C. Djeraba (1998). Visual content based retrieval in an image database with relevant feedback. In 'Multi-Media Database Management Systems, 1998. Proceedings. International Workshop on'. pp. 98 –105.

Brin, S. (1995). Near neighbor search in large metric spaces. In 'Very Large Data Bases (VLDB)'.

Brown, E., J.P. Callan and W.B. Croft (1994). Fast incremental indexing for full-text information retrieval. In 'Proceedings of the 20th International Conference on Very Large Databases (VLDB)'. Santiago, Chille. pp. 192 – 202.

Bruls, D., C. Huizing and J.J. Van Wijk. (2000). Squarified treemaps. In 'Data Visualization 2000, Proceedings of the Joint Eurographics and IEEE TCVG Symposium on Visualization'. Springer. Vienna. pp. 33–42.

Brunelli, R. and O. Mich (2000). 'Image retrieval by examples'. *IEEE Transactions on Multimedia* **2**(3), 164 –171.

Brunelli, R., O. Mich and C. M. Modena (1999). 'A survey on video indexing'. *Journal of Visual Communication and Image Representation* **10**, 78–112.

Bush, V. (1945). 'As we may think'. *The Atlantic Monthly.*

Carson, C. and Virginia E. Ogle (1996). 'Storage and retrieval of feature data for a very large online image collection'. *IEEE Computer Society Bulletin of the Technical Committee on Data Engineering* **19**(4), 19–27.

Carson, C., Serge Belongie, Hayit Greenspan and Jitendra Malik (1997). Region-based image querying. In 'Proceedings of the 1997 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'97)'. IEEE Computer Society. San Juan, Puerto Rico.

Catarci, T., Maria Francesca Costabile, Stefano Levialdi and Carlo Batini (1997). 'Visual query systems for databases: A survey'. *Journal of Visual Languages and Computing* **8**(2), 215–260.

Chang, E., James Ze Wang, Chen Li and Gio Wiederhold (1998). RIME: A replicated image detector for the world-wide web. In C.-C. J. Kuo, S.-F. Chang and S. Panchanathan (Eds.). 'Multimedia Storage and Archiving Systems III (VV02)'. Vol. 3527 of *SPIE Proceedings.* Boston, Massachusetts, USA. pp. 59–67. (SPIE Symposium on Voice, Video and Data Communications).

Chang, Y.-C., L. Bergmann, J. R. Smith and C.-S. Li (1999). Query taxonomy of multimedia databases. In Panchanathan et al. (Eds.). 'SPIE 23rd Symposium on Voice, Video and Data Communications'.

Chen, C. and Mary Czerwinski (1998). From latent semantics to spatial hypertext – an integrated approach. In 'Proceedings of the Ninth ACM Conference on Hypertext'. Mapping and Visualizing Navigation. pp. 77–86.

Chunyan, M. and Michael Junke Hu (1998). Querying and navigating of multimedia objects. In C.-C. J. Kuo, S.-F. Chang and S. Panchanathan (Eds.). 'Multimedia Storage and Archiving Systems III (VV02)'. Vol. 3527 of *SPIE Proceedings.* Boston, Massachusetts, USA. pp. 386–397. (SPIE Symposium on Voice, Video and Data Communications).

Comaniciu, D. and Peter Meer (1997). Robust analysis of feature spaces: Color image segmentation. In 'Proceedings of the 1997 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'97)'. IEEE Computer Society. San Juan, Puerto Rico. pp. 750–755.

Comaniciu, D., Peter Meer, Kin Xu and David Tyler (1999). Retrieval performance improvement through low rank corrections. In 'IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL'99)'. Fort Collins, Colorado, USA. pp. 50–54.

Corridoni, J. M., Alberto Del Bimbo and Enrico Vicario (1998). 'Image retrieval by color semantics with incomplete knowledge'. *Journal of the American Society for Information Science* **49**(3), 267–282.

Cover, T. M. and Joy A. Thomas (1991). *Elements of Information Theory.* John Wiley & Sons, Inc.

Cox, I. J., Matt L. Miller, Stephen M. Omohundro and Peter N. Yianilos (1996). Target testing and the `PicHunter` Bayesian multimedia retrieval system. In 'Advances in Digital Libraries (ADL'96)'. Library of Congress, Washington, D. C.. pp. 66–75.

Cox, I. J., Matthew L. Miller, Thomas P. Minka and Peter N. Yianilos (1998). An optimized interaction strategy for bayesian relevance feedback. In 'Proceedings of the 1998 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'98)'. Santa Barbara, California, USA. pp. 553–558.

Crevier, D. and Richard Lepage (Aug 1997). 'Knowledge-based image understanding systems: A survey'. *Computer Vision and Image Understanding* **67**(2), 161—185.

Das, M., R. Manmatha and Edward M. Riseman (1998). Indexing flowers by color names using domain knowledge-driven segmentation. In 'Proceedings of the Fourth IEEE Workshop on Applications of Computer Vision (WACV'98)'. Princeton, NJ, USA. pp. 94–99.

Deerwester, S., Susan T. Dumais, George W. Furnas, T. K. Landauer and Richard A. Harshman (1990). 'Indexing by latent semantic analysis'. *Journal of the American Society for Information Science* **41**(6), 391–407.

Degan, N. D., R. Lancini, P. Migliorati and S.Pozzi (1991). 'Efficient navigation in an image retrieval system'. *IEEE Globecom 91* **3 volumes**, 664–668.

Dimai, A. (1997). Spatial encoding using differences of global features. In I. K. Sethi and R. C. Jain (Eds.). 'Storage and Retrieval for Image and Video Databases V'. Vol. 3022 of *SPIE Proceedings*. pp. 352–360.

Ding, C. H. Q. (1999). A similarity-based probability model for latent semantic indexing. In 'Research and Development in Information Retrieval'. pp. 58–65.

Do, M. N. and Martin Vetterli (2002). 'Wavelet-based texture retrieval using generalized gaussian density and kullback-leibler distance'. *IEEE Transactions on Image Processing* **11**(2), 146 –158.

Dominich, S. (2000). 'A unified mathematical definition of classical information retrieval'. *Journal of the American Society for Information Science* **51**(7), 614–625.

Dotsenko, V. (1988). 'Neural networks: Translation-, rotation-, and scale-invariant pattern recognition'. *Journal of Physics A: Mathematical and General* **21**, L783–L787.

Dourish, P., W. Keith Edwards, Anthony LaMarca and Michael Salisbury (1999). 'Presto: An experimental architecture for fluid interactive document spaces'. *ACM SIGCHI Bulletin* **6**(2), 133–161.

Dumais, S. T., Todd A. Letsche, Michael L. Littman and Thomas K. Landauer (1997). Automatic cross-language retrieval using latent semantic indexing. In 'AAAI Symposium on CrossLanguage Text and Speech Retrieval'.

Dunn, G. (1989). *Design and analysis of reliability studies; the statistical evaluation of measurement errors*. Oxford University Press. 200 Madison Avenue, New York, NY 10016.

Duygulu, P., Kobus Barnard, Nando de Freitas and David Forsyth (2002). Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In 'European Conference on Computer Vision (ECCV) Copenhagen'. ECCV.

Exalibure Inc. (1998). 'Excalibur Visual RetrievalWare'. web page: `http://www.excalib.com/products/vrw/vrw.html`.

Excalibur Inc. (1997). 'Excalibur Visual RetrievalWare SDK 2.1 Technical summary'. web page: `http://www.excalib.com`.

Faloutsos, C. and Douglas W. Oard (1995). A survey of information retrieval and filtering methods. Technical Report CS-TR-3514. University of Maryland, College Park.

Faloutsos, C. et al. (1994). 'Efficient and effective querying by image content'. *Journal of Intelligent Information Systems* **3**, 231–262.

Faudemay, P., Gwenaël Durand, Claude Seyrat and Nicolas Tondre (1998). Indexing and retrieval of multimedia objects at different levels of granularity. In C.-C. J. Kuo, S.-F. Chang and S. Panchanathan (Eds.). 'Multimedia Storage and Archiving Systems III (VV02)'. Vol. 3527 of *SPIE Proceedings*. Boston, Massachusetts, USA. pp. 112–121. (SPIE Symposium on Voice, Video and Data Communications).

Flickner, M., Harpreet Sawhney, Wayne Niblack, Jonathon Ashley, Qian Huang, Byron Dom, Monika Gorkani, Jim Hafner, Denis Lee, Dragutin Petkovic, David Steele and Peter Yanker (1995). 'Query by image and video content: The QBIC system'. *IEEE Computer* **28**(9), 23–32.

Foltz, P. (1995). Improving human-proceedings interaction: Indexing the CHI index. In 'Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems'. Vol. 2 of *Interactive Posters*. pp. 101–102.

Foltz, P. W. (1990). Using latent semantic indexing for information filtering. In R. B. Allen (Ed.). 'Proceedings of the Office Information Systems Conference'. Filtering, Querying, and Navigating. Cambridge, MA. p. 40.

Frese, T., Charles A. Bouman and Jan P. Allebach (1997). Methodology for designing image similarity metrics based on human visual system models. In B. E. Rogowitz and T. N. Pappas (Eds.). 'Human Vision and Electronic Imaging II'. Vol. 3016 of *SPIE Proceedings*. San Jose, CA, USA. pp. 472–483.

Frost, C. O., Bradley Taylor, Anna Noakes, Stephen Markel, Deborah Torres and Karen M. Drabenstott (2000). 'Browse and search patterns in a digital image database'. *Information Retrieval* **1**(4), 287–313.

Gargi, U. and Rangachar Kasturi (1999). Image database querying using a multi-scale localized color representation. In 'IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL'99)'. Fort Collins, Colorado, USA. pp. 28–32.

Geman, D. and Roland Moquet (1999). A stochastic feedback model for image retrieval. Technical report. Ecole Polytechnique. 91128 Palaiseau Cedex, France.

Gevers, T. and A. W. M. Smeulders (1996). A comparative study of several color models for color image invariant retrieval. In A. W. M. Smeulders and R. Jain (Eds.). 'Image Databases and Multi-Media Search'. Intelligent Sensory Information Systems, Faculty of Mathematics, Computer Science, Physics and Astronomy. Amsterdam University Press. Kruislaan 403, 1098 SJ Amsterdam, The Netherlands. pp. 17–26.

Gevers, T. and A. W. M. Smeulders (1997). Object recognition based on photometric color invariants. In M. Frydrych, J. Parkkinen and A. Visa (Eds.). 'The 10th Scandinavian Conference on Image Analysis (SCIA'97)'. Pattern Recognition Society of Finland. Lappeenranta, Finland. pp. 861–868.

Golub, G. and C. Reinsch (1971). *Handbook for automatic computation II, linear algebra.* Springer-Verlag.

Golub, G. and C. van Loan (1983). *Matrix Computations.* Johns-Hopkins.

Gordon, A. S. (2000). Using annotated video as an information retrieval interface. In 'Proceedings of the 2000 International Conference on Intelligent User Interfaces'. pp. 133–140.

Gotoh, Y. and Steve Renals (1997). Document space models using latent semantic analysis. In 'Proc. Eurospeech '97'. Rhodes, Greece. pp. 1443–1446.

Gray, R. S. (1995). Content-based image retrieval: color and edges. Technical Report PCS-TR95-252. Dartmouth College, Computer Science. Hanover, NH.

Greiff, W. R. (1998). A theory of term weighting based on exploratory data analysis. In W. B. Croft, A. Moffat, C. J. van Rijsbergen, R. Wilkinson and J. Zobel (Eds.). 'Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval'. ACM Press, New York. Melbourne, Australia. pp. 11–19.

Grossberg, S. (1980). 'How does a brain build a cognitive code?'. *Psychological Review* **87**, 1–51.

Gupta, A. and Ramesh Jain (1997). 'Visual information retrieval'. *Communications of the ACM* **40**(5), 70–79.

Gupta, L. and M.D. Srinath (1987). 'Contour sequence moments for the classification of closed planar shapes'. *Pattern Recognition* **20**(3), 267—272.

Hafner, J., Harpreet S. Sawhney, Will Equitz, Myron Flickner and Wayne Niblack (1995). 'Efficient color histogram indexing for quadratic form distance functions'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **17**(7), 729–736.

Han, K. and S.-H. Myaeng (1996). Image organization and retrieval with automatically constructed feature vectors. In H.-P. Frei, D. Harman, P. Schäuble and R. Wilkinson (Eds.). 'Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'96)'. Zürich, Switzerland. pp. 157–165.

Havaldar, P., Gérard Medioni and Fridtjof Stein (1996). 'Perceptual grouping for generic recognition'. *International Journal of Computer Vision* **20**(1/2), 59–80.

Hearst, M. (1999). *Modern Information Retrieval.* paperback edn. Addison-Wesley. chapter 10 : User Interfaces and Visualization, pp. 257–322.

Heeger, D. J. and James R. Bergen (1995). Pyramid based texture analysis/synthesis. In 'Proceedings of SGIGRAPH'95'. Los Angeles, CA, USA. pp. 229–238.

Hendrix, G. (1978). Semantic knowledge. In D. Walker (Ed.). 'Understanding Spoken Language'. North Holland. New York.

Hirata, K. and T. Kato (1992). Query by visual example. In 'EDBT'92'. pp. 56–71.

Hofmann, T. (1999*a*). Probabilistic latent semantic analysis. In K. B. Laskey and H. Prade (Eds.). 'Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI-99)'. Morgan Kaufmann Publishers. S.F., Cal.. pp. 289–296.

Hofmann, T. (1999*b*). Probabilistic latent semantic indexing. In 'SIGIR Conference'. pp. 50–57.

Horn, R. A. and Charles R. Johnson (1991). *Topics in Matrix Analysis.* Cambridge University Press.

Hsu, W., T.S. Chua and H.K. Pung (2000). 'Approximating content-based object-level image retrieval'. *Multimedia Tools and Applications* **12**(1), 59–79.

Huang, J., S. Ravi Kumar and Mandar Mitra (1997*a*). Combining supervised learning with color correlograms for content-based image retrieval. In 'Proceedings of The Fifth ACM International Multimedia Conference (ACM Multimedia 97)'. Seattle, WA, USA. pp. 325–334.

Huang, J., S. Ravi Kumar, Mandar Mitra, Wei-Jing Zhu and Ramin Zabih (1997*b*). Image indexing using color correlograms. In 'Proceedings of the 1997 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'97)'. IEEE Computer Society. San Juan, Puerto Rico. pp. 762–768.

Huet, B. and Edwin R. Hancock (1999). Inexact graph retrieval. In 'IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL'99)'. Fort Collins, Colorado, USA. pp. 40–44.

Hull, D. (1994). Improving text retrieval for the routing problem using latent semantic indexing. In W. B. Croft and C. J. van Rijsbergen (Eds.). 'Proceedings of the 17th Annual International Conference on Research and Development in Information Retrieval'. Springer Verlag. London, UK. pp. 282–291.

Hwang, W.-S., John J. Weng, Ming Fang and Jianzhong Qian (1999). A fast image retrieval algorithm with automatically extracted discriminant features. In 'IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL'99)'. Fort Collins, Colorado, USA. pp. 8–12.

I^{3}A (2002). 'Internet imaging protocol'. Web page : `http://www.i3a.org/i_iip.html`.

IBM (1998). 'QBIC^{TM} - IBM's Query By Image Content'. web page: `http://wwwqbic.almaden.ibm.com/~qbic/`.

Inder, R. and J. Stader (1995). Sustaining interaction in database query. In 'Proceedings of HCI International'. Yokohama, Japan. pp. 711–716.

Iqbal, Q. and J. K. Aggarwal (1999). Applying perceptual grouping to content-based image retrieval: Building images. In 'Proceedings of the 1999 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'99)'. IEEE Computer Society. Fort Collins, Colorado, USA. pp. 42–48.

Ishikawa, Y., Ravishankar Subramanya and Christos Faloutsos (1998). Mindreader: Querying databases through multiple examples. In 'Proceedings of 24th International Conference on Very Large Databases (VLDB'98)'. New York, NY, USA. pp. 218–227.

ISO/IEC (2002). 'MPEG-7 standard'. Web page : `http://mpeg.telecomitalialab.com/standards/mpeg-7/mpeg-7.htı`

Iyengar, G. and Andrew B. Lippman (1998). Semantically controlled contentbased retrieval of video sequences. In C.-C. J. Kuo, S.-F. Chang and S. Panchanathan (Eds.). 'Multimedia Storage and Archiving Systems III (VV02)'. Vol. 3527 of *SPIE Proceedings*. Boston, Massachusetts, USA. pp. 223–232. (SPIE Symposium on Voice, Video and Data Communications).

Jacobs, C. E., Adam Finkelstein and David H. Salesin (1995). Fast multiresolution image querying. In 'Proceedings of SIGGRAPH 95 (Los Angeles, California, August 6–11, 1995)'. ACM. New York.

Jain, A. and G. Healey (1998). 'A multiscale representation including opponent color features for texture recognition'. *IEEE Transactions on Image Processing* **7**(1), 124–128.

Jain, A. K. (1989). *Fundamentals of digital image processing.* Prentice-Hall information and system sciences series. Prentice-Hall International. London.

Jain, A. K. and Aditya Vailaya (1996). 'Image retrieval using color and shape'. *Pattern Recognition* **29**(8), 1233–1244.

Jianbo Shi; Malik, J. (2000). 'Normalized cuts and image segmentation'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(8), 888 -905.

Jin, J. S., Ruth Kurniawati, Guangyu Xu and Xuesheng Bai (1998). Using browsing to improve content-based image retrieval. In C.-C. J. Kuo, S.-F. Chang and S. Panchanathan (Eds.). 'Multimedia Storage and Archiving Systems III (VV02)'. Vol. 3527 of *SPIE Proceedings*. Boston, Massachusetts, USA. pp. 101–109. (SPIE Symposium on Voice, Video and Data Communications).

Johnson, B. and Ben Shneiderman (1991). Treemaps: A space-filling approach to the visualization of hierarchical information structures. In 'Proc. Of the 2nd International IEEE Visualization Conference'. pp. 284–291.

Jones, K. S. (1991). 'The role of artificial intelligence in information retrieval'. *Journal of the American Society for Information Science* **42**(8), 558–565.

J.W. Sammon, J. (1969). 'A nonlinear mapping for data structure analysis'. *IEEE Transactions on Computers* **C-18**, 401–409.

Kato, T. (1992). Database architecture for content-based image retrieval. In A. A. Jamberdino and W. Niblack (Eds.). 'Image Storage and Retrieval Systems'. Vol. 1662 of *SPIE Proceedings*. San Jose, California. pp. 112–123.

Kauniskangas, H., J. Sauvola, M. Pietkäinen and D. Doermann (1997). Content-based image retrieval using composite features. In M. Frydrych, J. Parkkinen and A. Visa (Eds.). 'The 10th Scandinavian Conference on Image Analysis (SCIA'97)'. Pattern Recognition Society of Finland. Lappeenranta, Finland. pp. 35–42.

Kelly, P. M., Michael Cannon and Donald R. Hush (1995). Query by image example: the CANDID approach. In W. Niblack and R. C. Jain (Eds.). 'Storage and Retrieval for Image and Video Databases III'. Vol. 2420 of *SPIE Proceedings*. pp. 238–248.

Kim, Y. H., K. E. Lee, K. S. Choi, J. H. Yoo, P. K. Rhee and Y. C. Park (1998). Personalized image retrieval with user's preference model. In C.-C. J. Kuo, S.-F. Chang and S. Panchanathan (Eds.). 'Multimedia Storage and Archiving Systems III (VV02)'. Vol. 3527 of *SPIE Proceedings*. Boston, Massachusetts, USA. pp. 47–55. (SPIE Symposium on Voice, Video and Data Communications).

Kohonen, T., Jussi Hynninena, Jari Kangas and Jorma Laaksonen (1996). SOM_PAK: The self-organizing map program package. Technical Report A31. Helsinki University of Technology, Laboratory of Computer and Information Science. Finland.

Kohonen, T., T. S. Huang (ed.) and M. R. Schroeder (ed.) (2001). *Self-Organizing Maps*. Vol. 30 of *Springer Series in Information Sciences*. 3 edn. Springer Verlag.

Kurimo, M. (1999). Indexing audio documents by using latent semantic analysis and SOM. In E. Oja and S. Kaski (Eds.). 'Kohonen Maps'. Elsevier. Amsterdam. pp. 363–374. Keywords: audio indexing, latent semantic analysis, self-organising map, speech recognition, information retrieval.

Kurimo, M. (2000). Fast latent semantic indexing of spoken documents by using self-organizing maps. In '2000 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '00'. Vol. 4. pp. 2425–2428.

La Cascia, M., S. Sethi and S. Sclaroff (1998). Combining textual and visual cues for content-based image retrieval on the world-wide web. In 'Proc. of the IEEE Workshop on Content-based Access of Image and Video Libraries'. pp. 24–28.

Laurini, R. (Ed.) (2000). *Advances in Visual Information Systems, 4th International Conference, VISUAL 2000, Lyon, France, November 2-4, 2000, Proceedings*. Vol. 1929 of *Lecture Notes in Computer Science*. Springer.

Lazarsfeld, P. F. and N. W. Henry (1968). *Latent Structure Analysis*. Houghton Mifflin. New York.

Leake, D. B. and Ryan Scherle (2001). Towards context-based search engine selection. In 'Proceedings of the 2001 International Conference on Intelligent User Interfaces'. pp. 109–112.

Lew, M. S., D. P. Huijsmans and Dee Denteneer (1996). Content based image retrieval: KLT, projections, or templates. In A. W. M. Smeulders and R. Jain (Eds.). 'Image Databases and Multi-Media Search'. Intelligent Sensory Information Systems, Faculty of Mathematics, Computer Science, Physics and Astronomy. Amsterdam University Press. Kruislaan 403, 1098 SJ Amsterdam, The Netherlands. pp. 27–34.

Li, S. (1992). 'Matching: Invariant to translations, rotations and scale changes'. *Pattern Recognition* **25**(6), 583–594.

Ling, C. and S.F. Chang (1998). A robust image authentification method distinguishing JPEG compression from maliscious manipulation. In 'SPIE: Storage and Retrieval of Image/Video Databases'. San Jose.

Liu, F. and R.W. Picard (1996). 'Periodicity, directionality, and randomness: Wold features for image modeling and retrieval'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**(7), 722–733.

Loncaric, S. (1998). 'A survey of shape analysis techniques'. *Pattern Recognition* **31**(8), 983–1001.

Lucchese, L. and Sanjit K. Mitra (1999*a*). Unsupervised segmentation of color images based on k-means clustering in the chromaticity plane. In 'IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL'99)'. Fort Collins, Colorado, USA. pp. 74–78.

Lucchese, L. and S.K. Mitra (1999*b*). Advances in color image segmentation. In 'Global Telecommunications Conference, GLOBECOM '99'. pp. (vol. 4) 2038 –2044.

Luhn, H. (1957). 'A statistical approach to mechanised encoding and searching of library information'. *IBM Journal of Research and Development* **1**, 309–317.

Ma, W. Y. and B. S. Manjunath (1998). 'A texture thesaurus for browsing large aerial photographs'. *Journal of the American Society for Information Science* **49(7)**, 633–648.

Ma, W. Y., Yining Deng and B. S. Manjunath (1997). Tools for texture- and color-based search of images. In B. E. Rogowitz and T. N. Pappas (Eds.). 'Human Vision and Electronic Imaging II'. Vol. 3016 of *SPIE Proceedings*. San Jose, CA. pp. 496–507.

MacLennan, B. (1991). Characteristics of connectionist knowledge representation. Technical Report CS-91-147. Computer Science Department, University of Tennessee, Knoxville.

Maillet, S. M. et al. (2000—). 'Multimedia retrieval markup language web site'. http://www.mrml.net.

Maletic, J. I. and N. Valluri (1999). Automatic software clustering via latent semantic analysis. In '14th IEEE International Conference on Automated Software Engineering'. IEEE Computer Society Press. pp. 251–254.

Malik, J. (2001). Visual grouping and object recognition. In 'Proceedings of the 11th International Conference on Image Analysis and Processing'. IEEE. pp. 612 –621.

Manjunath, B., J.-R. Ohm, V.V. Vasudevan and A. Yamada (2001). 'Color and texture descriptors'. *IEEE Transactions on Circuits and Systems for Video Technology* **11**(6), 703 –715.

Mao, J. and A. K. Jain (1995). 'Artificial neural networks for feature extraction and multivariate data projection'. *IEEE Trans. on Neural Networks* **6**, 296–317.

Marchand-Maillet, S. (1997—). 'Viper project'. http://viper.unige.ch.

Markkula, M. and Eero Sormunen (1998). Searching for photos - journalists' practices in pictorial IR. In J. P. Eakins, D. J. Harper and J. Jose (Eds.). 'The Challenge of Image Retrieval, A Workshop and Symposium on Image Retrieval'. Electronic Workshops in Computing. The British Computer Society. Newcastle upon Tyne.

Markkula, M. and Eero Sormunen (1999). 'End-user searching challenges indexing practices in the digital photo archive'. *Information Retrieval.* (to appear).

Markkula, M., Marius Tico, Bemmu Sepponen, Katja Nirkkonen and Eero Sormunen (2001). 'A test collection for the evaluation of content-based image retrieval algorithms — a user and task-based approach'. *Information Retrieval* **4**(3/4), 275–293.

McCabe, A., Geoff West and Terry Caelli (1998). Filter techniques for complex spatio-chromatic image processing. In 'International Conference on Image Processing'. Chicago, Il., USA.

Meghini, C., Fabrizio Sebastiani and Umberto Straccia (2001). 'A model of multimedia information retrieval'. *Journal of the ACM* **48**(5), 909–970.

Minka, T. (1996). An image database browser that learns from user interaction. Master's thesis. MIT Media Laboratory. 20 Ames St., Cambridge, MA 02139.

Minka, T. and R.W. Piccard (1996). A society of models for video and image libraries. Technical Report 349. M.I.T. Media Laboratory Perceptual Computing Section.

Mohan, R., John R. Smith and Chung-Sheng Li (1998). Multimedia content customization for universal access. In C.-C. J. Kuo, S.-F. Chang and S. Panchanathan (Eds.). 'Multimedia Storage and Archiving Systems III (VV02)'. Vol. 3527 of *SPIE Proceedings*. Boston, Massachusetts, USA. pp. 410–418. (SPIE Symposium on Voice, Video and Data Communications).

Mokhtarian, F., Sadegh Abbasi and Josef Kittler (1996). Efficient and robust retrieval by shape content through curvature scale space. In A. W. M. Smeulders and R. Jain (Eds.). 'Image Databases and Multi-Media Search'. Intelligent Sensory Information Systems, Faculty of Mathematics, Computer Science, Physics and Astronomy. Amsterdam University Press. Kruislaan 403, 1098 SJ Amsterdam, The Netherlands. pp. 35–42.

Müller, H., David McG. Squire, Wolfgang Müller and Thierry Pun (1999). Efficient access methods for content-based image retrieval with inverted files. In S. Panchanathan, S.-F. Chang and C.-C. J. Kuo (Eds.). 'Multimedia Storage and Archiving Systems IV (VV02)'. Vol. 3846 of *SPIE Proceedings*. Boston, Massachusetts, USA. (SPIE Symposium on Voice, Video and Data Communications).

Müller, W., Henning Müller, Stéphane Marchand-Maillet, Thierry Pun, David McG. Squire, Zoran Pecenovic, Christoph Giess and Arjen P. de Vries (2000*a*). Mrml: A communication protocol for content-based image retrieval. In 'International Conference on Visual Information Systems (Visual 2000)'. Lyon, France.

Müller, W., Zoran Pecenovic, Henning Müller, Stephane Marchand-Maillet, Thierry Pun, David Squire, Arjen P. De Vries and Christoph Giess (2000*b*). Mrml: An extensible communication protocol for interoperability and benchmarking of multimedia information retrieval systems. In 'SPIE Photonics East - Voice, Video, and Data Communications'. Boston, MA, USA.

Niblack, W. (1999). SlideFinder: A tool for browsing presentation graphics using content-based retrieval. In 'IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL'99)'. Fort Collins, Colorado, USA. pp. 114–118.

Niblack, W., Ron Barber, Will Equitz, Myron D. Flickner, Eduardo H. Glasman, Dragutin Petkovic, Peter Yanker, C. Faloutsos and Gabriel Taubin (1993). QBIC project: querying images by content, using color, texture, and shape. In W. Niblack (Ed.). 'Storage and Retrieval for Image and Video Databases'. Vol. 1908 of *SPIE Proceedings*. pp. 173–187.

Nielsen, J. (1993). *Usability Engineering*. Academic Press. Boston, MA.

Oard, D. W. (2000). 'User interface design for speech-based retrieval'. *Bulletin of The American Society for Information Science*.

O'Brian, G. W. (1994). Information management tools for updating an SVD-encoded indexing scheme. Master's thesis. The University of Knoxville. Knoxville Tennessee.

Ogle, V. E. and Michael Stonebraker (1995). 'Chabot: Retrieval from a relational database of images'. *IEEE Computer*.

Ortega, M., Yong Rui, Kaushik Chakrabarti, Kriengkrai Porkaew, Sharad Mehrotra and Thomas S. Huang (1998). 'Supporting ranked boolean similarity queries in MARS'. *IEEE Transactions on Knowledge and Data Engineering*.

Ovaska, P. and Jussi Parkkinen (1997). A pictorial object-oriented database architecture for retrieving images by their content. In M. Frydrych, J. Parkkinen and A. Visa (Eds.). 'The 10th Scandinavian Conference on Image Analysis (SCIA'97)'. Pattern Recognition Society of Finland. Lappeenranta, Finland. pp. 59–64.

Ozer, B., Wayne Wolf and Ali N. Akansu (1999). A graph based object description for information retrieval in digital image and video libraries. In 'IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL'99)'. Fort Collins, Colorado, USA. pp. 79–83.

Pal, N. and S. Pal (1993). 'A review on image segmentation techniques'. *Pattern Recognition* **26**(9), 1277—1294.

Papadimitriou, C. H., Hisao Tamaki, Prabhakar Raghavan and Santosh Vempala (1998). Latent semantic indexing: A probabilistic analysis. In ACM (Ed.). 'PODS '98. Proceedings of the ACM SIGACT–SIGMOD–SIGART Symposium on Principles of Database Systems, June 1–3, 1998, Seattle, Washington'. ACM Press. New York, NY 10036, USA. pp. 159–168.

Pass, G. and Ramin Zabih (1996). Histogram refinement for content-based image retrieval. In 'IEEE Workshop on Applications of Computer Vision'. Sarasota, Florida, USA. pp. 96–102.

Paynter, G. W., Ian H. Witten, Sally Jo Cunningham and George Buchanan (2000). Scalable browsing for large collections: A case study. In 'Proceedings of the 5th ACM International Conference on Digital Libraries'. pp. 215–223.

Pecenovic, Z. (1997). Image retrieval using latent semantic indexing. Final year graduate thesis. AudioVisual Communications Lab, Ecole Polytechnique Fédérale de Lausanne. Switzerland.

Pecenovic, Z. (1998). Finding rainbows on the internet. Master's thesis. EPFL.

Pecenovic, Z. and Pearl Pu (2000). Dynamic overview techniques for image retrieval. In 'VisSym'00,Second Joint Eurographics-IEEE TCVG Symposium on Visualization'. Amsterdam, The Netherlands.

Pecenovic, Z., Minh Do, Martin Vetterli and Pearl Pu (2000). Integrated browsing and searching of large image collections. In 'International Conference on Visual Information Systems (Visual 2000)'. Lyon, France.

Pecenovic, Z., Minh Do, Serge Ayer and Martin Vetterli (1998). New methods for image retrieval. In 'Proceedings of the International Congress on Imaging Science'. Vol. 2. University of Antwerp, Belgium. pp. 242–246.

Pecenovic, Z., Serge Ayer and Martin Vetterli (2001). Joint textual and visual cues for retrieving images using latent semantic indexing. In 'Proceedings of the International Workshop on Content-Based Multimedia Indexing'.

Pentland, A. (1976). Classification by clustering. In 'IEEE Proc. Symp. on Machine Processing of Remotely Sensed Data'. Purdue, Indiana.

Pentland, A., R. W. Picard and S. Sclaroff (1996). 'Photobook: Tools for content-based manipulation of image databases'. *International Journal of Computer Vision* **18**(3), 233–254.

Picard, R. W. (1995). Toward a visual thesaurus. Technical Report 358. MIT Media Laboratory Perceptual Computing Section. 20 Ames St., Cambridge MA 02139.

Picard, R. W., Thomas P. Minka and Martin Szummer (1996). Modeling user subjectivity in image libraries. In P. Delogne (Ed.). 'IEEE International Conference on Image Processing (ICIP'96)'. Lausanne, Switzerland.

Porkaew, K., Michael Ortega and Sharad Mehrotra (1999). Query reformulation for content based multimedia retrieval in MARS. In 'ICMCS, Vol. 2'. pp. 747–751.

Porter, M. (1980). 'An algorithm for suffix stripping'. *Program* **14**(3), 130–137.

Portilla, J. and Eero P. Simoncelli (1999). Texture representation and synthesis using correlation of complex wavelet coefficient magnitudes. Technical Report 54. Consejo Superior de Investigaciones Cientificas (CSIC). Madrid.

Pratt, W. K. (1991). *Digital Image Processing.* 2 edn. John Wiley & Sons, Inc. New York.

Puzicha, J., Y. Rubner, C. Tomasi and J. Buhmann (1999). Empirical evaluation of dissimilarity measures for color and texture. In 'Proceedings the IEEE International Conference on Computer Vision(ICCV-1999)'. pp. 1165–1173.

Quintana, Y. (1997). Organization and retrieval in a pictorial digital library. In R. B. Allen and E. Rasmussen (Eds.). 'Proceedings of the 2nd ACM International Conference on Digital Libraries (ACMDL'97)'. Association for Computing Machinery. Philadelphia, PA. pp. 13–20.

Randen, T. and J.H. Husoy (1999). 'Filtering for texture classification: A comparative study'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **21**(4), 291 –310.

Ravela, S. and R. Manmatha (1998). On computing global similarity in images. In 'Proceedings of the Fourth IEEE Workshop on Applications of Computer Vision (WACV'98)'. Princeton, NJ, USA. pp. 82–87.

Ritter, H., Thomas Martinez and Klaus Schulten (1992). *Neural computation and self-organizing maps: an introduction.* Computation and neural systems series. Addison-Wesley Publishing Company.

Ro, Y. M. (1998). Matching pursuit: Contents featuring for image indexing. In C.-C. J. Kuo, S.-F. Chang and S. Panchanathan (Eds.). 'Multimedia Storage and Archiving Systems III (VV02)'. Vol. 3527 of *SPIE Proceedings.* Boston, Massachusetts, USA. pp. 89–100. (SPIE Symposium on Voice, Video and Data Communications).

Rogowitz, B. E., Thomas Frese, John R. Smith, Charles A. Bouman and Edward B. Kalin (1998). Perceptual image similarity experiments. In B. E. Rogowitz and T. N. Pappas (Eds.). 'Human Vision and Electronic Imaging III'. Vol. 3299 of *SPIE Proceedings.* San Jose, CA, USA. pp. 576–590.

Rubner, Y., Carlo Tomasi and Leonidas Guibas (1998). A metric for distributions with applications to image databases. In 'Proceedings of IEEE Int. Conf. On Computer Vision'.

Rubner, Y., Leonidas Guibas and Carlo Tomasi (1997). The earth mover's distance, multi-dimensional scaling, and color-based image retrieval. In 'Proceedings of the ARPA Image Understanding Workshop'.

Rui, Y., Thomas S. Huang and Sharad Mehrotra (1997). Relevance feedback techniques in interactive content-based image retrieval. In I. K. Sethi and R. C. Jain (Eds.). 'Storage and Retrieval for Image and Video Databases VI'. Vol. 3312 of *SPIE Proceedings.* pp. 25–36.

Rui, Y., Thomas S. Huang and Shih-Fu Chang (1999). 'Image retrieval: Current techniques, promising directions and open issues'. *Journal of Visual Communication and Image Representation* **10**(4), 39–62.

Rui, Y., Thomas S. Huang, Michael Ortega and Sharad Mehrotra (1998). 'Relevance feedback: A power tool in interactive content-based image retrieval'. *IEEE Transactions on Circuits and Systems for Video Technology* **8**(5), 644–655. (Special Issue on Segmentation, Description, and Retrieval of Video Content).

Ruthven, I. (2000). 'Incorporating aspects of information use into relevance feedback'. *Information Retrieval* **2**(1), 83–88.

Salton, G. and Chris Buckley (1988). 'Term weighting approaches in automatic text retrieval'. *Information Processing and Management* **24**(5), 513–523.

Santini, S. and Ramesh Jain (1996). Similarity queries in image databases. In 'Proceedings of the 1996 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'96)'. San Francisco, California. pp. 646–651.

Schonfeld, D. and Dan Lelescu (1998). VORTEX: Video retrieval and tracking from compressed multimedia databases – Template matching from MPEG2 video compression standard. In C.-C. J. Kuo, S.-F. Chang and S. Panchanathan (Eds.). 'Multimedia Storage and Archiving Systems III (VV02)'. Vol. 3527 of *SPIE Proceedings.* Boston, Massachusetts, USA. pp. 233–244. (SPIE Symposium on Voice, Video and Data Communications).

Sclaroff, S. (1995). World wide web image search engines. Technical Report 95-016. Computer Science Department, Boston University. 111 Cummington St., Boston, MA, USA.

Sclaroff, S., Leonid Taycher and Marco La Cascia (1997). ImageRover: a content-based browser for the world wide web. In 'IEEE Workshop on Content-Based Access of Image and Video Libraries'. San Juan, Puerto Rico. pp. 2–9.

Shi, J. and J. Malik (2000). 'Normalized cuts and image segmentation'. *PAMI* **22(8)**, 888–905.

Shneiderman, B. (1983). 'Direct manipulation: A step beyond programming languages'. *IEEE Computer* **16**, 57—69.

Shneiderman, B. (1994). 'Dynamic queries for visual information seeking'. *IEEE Software* **11**(6), 70–77.

Shneiderman, B. (1998). *Designing the User Interface: Strategies for Effective Human-Computer Interaction.* 3 edn. Addison-Wesley. Reading, MA.

Shneiderman, B. (2002). 'Treemaps'. Web page : `http://www.cs.umd.edu/hcil/treemaps/`.

Shneiderman, B., David Feldman, Anne Rose and Xavier Ferre Grau (2000). Visualizing digital library search results with categorical and hierarchical axes. In 'Proceedings of the 5th ACM International Conference on Digital Libraries'. pp. 57–66.

Smeulders, A., M. Worring, S. Santini, A. Gupta and R. Jain (2000). 'Content-based image retrieval at the end of the early years'. *Pattern Analysis and Machine Intelligence* **22**(12), 1349 –1380.

Smith, J. and Shih-Fu Chang (1997*a*). 'Visually searching the web for content'. *IEEE Multimedia* **4(3)**, 12 –20.

Smith, J. R. and Shih-Fu Chang (1996*a*). Tools and techniques for color image retrieval. In I. K. Sethi and R. C. Jain (Eds.). 'Storage & Retrieval for Image and Video Databases IV'. Vol. 2670 of *IS&T/SPIE Proceedings.* San Jose, CA, USA. pp. 426–437.

Smith, J. R. and Shih-Fu Chang (1996*b*). VisualSEEk: a fully automated content-based image query system. In 'The Fourth ACM International Multimedia Conference and Exhibition'. Boston, MA, USA.

Smith, J. R. and Shih-Fu Chang (1997*b*). Enhancing image search engines in visual information environments. In 'Electronic Proceedings of the IEEE Signal Processing Society Workshop on Multimedia Signal Processing'. Princeton, New Jersey, USA.

Smith, J. R. and Shih-Fu Chang (1997*c*). Querying by color regions using the *VisualSEEk* content-based visual query system. In M. T. Maybury (Ed.). 'Proceedings of the IJCAI Workshop on Intelligent Multimedia Information Retrieval'.

Sormunen, E., Marjo Markkula and Kalervo Järvelin (1999). The perceived similarity of photos – seeking a solid basis for the evaluation of content-based retrieval algorithms. In 'Final Mira Conference'. Electronic Workshops in Computing. The British Computer Society. Glasgow.

Spark Jones, K. and Peter Willett (1997). *Readings in information retrieval.* Morgan Kaufmann Publishers, Inc.

Squire, D. M. (1998). Using human partitionings of image sets to learn a similarity-based distance measure for the organization of image databases. In C.-C. J. Kuo, S.-F. Chang and S. Panchanathan (Eds.). 'Multimedia Storage and Archiving Systems III (VV02)'. Vol. 3527 of *SPIE Proceedings.* Boston, Massachusetts, USA. pp. 80–88. (SPIE Symposium on Voice, Video and Data Communications).

Squire, D. M. and Thierry Pun (1997). A comparison of human and machine assessments of image similarity for the organization of image databases. In M. Frydrych, J. Parkkinen and A. Visa (Eds.). 'The 10th Scandinavian Conference on Image Analysis (SCIA'97)'. Pattern Recognition Society of Finland. Lappeenranta, Finland. pp. 51–58.

Squire, D. M. and Thierry Pun (1998). 'Assessing agreement between human and machine clusterings of image databases'. *Pattern Recognition* **31**(12), 1905–1919.

Squire, D. M., Wolfgang Müller and Henning Müller (1999*a*). Relevance feedback and term weighting schemes for content-based image retrieval. In D. P. Huijsmans and A. W. M. Smeulders (Eds.). 'Third International Conference On Visual Information Systems (VISUAL'99)'. number 1614 In 'Lecture Notes in Computer Science'. Springer-Verlag. Amsterdam, The Netherlands. pp. 549–556.

Squire, D. M., Wolfgang Müller, Henning Müller and Jilali Raki (1999*b*). Content-based query of image databases, inspirations from text retrieval: inverted files, frequency-based weights and relevance feedback. In 'The 10th Scandinavian Conference on Image Analysis (SCIA'99)'. Kangerlussuaq, Greenland. pp. 143–149.

Srihari, R. (1995*a*). 'Automatic indexing and content-based retrieval of captioned images'. *Computer* **28(9)**, 49 –56.

Srihari, R. (1995*b*). Combining text and image information in content-based retrieval. In 'ICIP 1995, International Conference on Image Processing'. Vol. 1. pp. 326 –329.

Stewart, B. S., Ching-Fang Liaw and Chelsea C. White III (1994). 'A bibliography of heuristic search through 1992'. *IEEE Transactions on Systems, Man and Cybernetics* **24**(2), 268–293.

Story, R. (1996). 'An explanation of the effectiveness of latent semantic indexing by means of a Bayesian regression model'. *Information Processing & Management* **32**(3), 329–344.

Stricker, M. and A. Dimai (1996). Color indexing with weak spatial constraints. In 'Storage and Retrieval for Image and Video Databases IV'. Vol. 2670 of *SPIE*. pp. 29–40.

Stricker, M. and M. Orengo (1995). Similarity of color images. In 'Storage and Retrieval for Image and Video Databases III'. Vol. 2420 of *SPIE*. pp. 381–392.

Swain, M. J. and Dana H. Ballard (1990). Indexing via color histograms. In 'Proceedings of the DARPA Image Understanding Workshop'. Pittsburgh, PA, USA. pp. 623–630.

Swain, M. J. and Dana H. Ballard (1991). 'Color indexing'. *International Journal of Computer Vision* **7**(1), 11–32.

Swanberg, D., Fe Shu Chiao and Ramesh Jain (1993). Architecture of a multimedia information system for content-based retrieval. In '?'. Vol. 712 of *Lecture Notes in Computer Science*. Kluwer Accademic Press. p. 387 ff.

Tague-Sutcliffe, J. (1997). The pragmatics of information retrieval experimentation, revisited. In K. Spark Jones and P. Willett (Eds.). 'Readings in Information Retrieval'. Multimedia Information and Systems. Morgan Kaufmann. 340 Pine Street, San Francisco, USA. chapter 4, pp. 205–216.

Tao, Y. and W. I. Grosky (2000). 'Image indexing and retrieval using object-based point feature maps'. *Journal of Visual Languages and Computing* **11**, 323—343.

Tao, Y. and W.I. Grosky (2001). 'Spatial color indexing using rotation, translation, and scale invariant anglograms'. *Multimedia Tools and Applications* **15**(3), 247–268.

Tappert, C., C.Y. Suen and T. Wakahara (1990). 'The state of the art in online handwriting recognition'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **12**(8), 787 –808.

Taubes (1995). 'Indexing the internet'. *SCIENCE: Science.*

Thomasian, A., Vittorio Castelli and Chung-Sheng Li (1998). CSVD: Approximate similarity searches in high dimensional spaces using clustering and singular value decomposition. In C.-C. J. Kuo, S.-F. Chang and S. Panchanathan (Eds.). 'Multimedia Storage and Archiving Systems III (VV02)'. Vol. 3527 of *SPIE Proceedings*. Boston, Massachusetts, USA. pp. 144–154. (SPIE Symposium on Voice, Video and Data Communications).

van Bakel, J. (1984). *Automatic Semantic Interpretation: A Computer Model of Understanding Language*. Foris. Dordrecht.

van Doorn, M. G. L. M. and Arjen P. de Vries (2000). The psychology of multimedia databases. In 'DL'00: Proceedings of the 5th ACM International Conference on Digital Libraries'. Full Papers. pp. 1–9.

Van Rijsbergen, C. J. (1979). *Information Retrieval.* 2 edn. Butterworth. London.

Vasconcelos, N. (2001). On the complexity of probabilistic image retrieval. In 'Proceedings. Eighth IEEE International Conference on Computer Vision'. IEEE. pp. (Vol II) 400 – 407.

Vasconcelos, N. and Andrew Lippman (1998*a*). A bayesian framework for content-based indexing and retrieval. In 'Data Compression Conference'. p. 580.

Vasconcelos, N. and Andrew Lippman (1998*b*). Embedded mixture modeling for efficient probabilistic contentbased indexing and retrieval. In C.-C. J. Kuo, S.-F. Chang and S. Panchanathan (Eds.). 'Multimedia Storage and Archiving Systems III (VV02)'. Vol. 3527 of *SPIE Proceedings*. Boston, Massachusetts, USA. pp. 134–143. (SPIE Symposium on Voice, Video and Data Communications).

Vasconcelos, N. and Andrew Lippman (1999). Probabilistic retrieval: new insights and experimental results. In 'IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL'99)'. Fort Collins, Colorado, USA. pp. 62–66.

Vellaikal, A. and C.-C. Jay Kuo (1995). Content-based image retrieval using multiresolution histogram representation. In C.-C. J. Kuo (Ed.). 'Digital Image Storage and Archiving Systems'. Vol. 2606 of *SPIE Proceedings*. Philadelphia, PA, USA. pp. 312–323.

Vellaikal, A. and C.-C. Jay Kuo (1998). Hierarchical clustering techniques for image database organization and summarization. In C.-C. J. Kuo, S.-F. Chang and S. Panchanathan (Eds.). 'Multimedia Storage and Archiving Systems III (VV02)'. Vol. 3527 of *SPIE Proceedings*. Boston, Massachusetts, USA. pp. 68–79. (SPIE Symposium on Voice, Video and Data Communications).

Veltkamp, R. C. and Mirele Tanase (2000). Content-based image retrieval systems: A survey. Technical Report UU-CS-2000-34. Department of Computing Science, Utrecht University.

Vendrig, J., M. Worring and A. W. M. Smeulders (1999). Filter image browsing: Exploiting interaction in image retrieval. In D. P. Huijsmans and A. W. M. Smeulders (Eds.). 'Third International Conference On Visual Information Systems (VISUAL'99)'. number 1614 In 'Lecture Notes in Computer Science'. Springer-Verlag. Amsterdam, The Netherlands. pp. 147–154.

Vendrig, J., Marcel Worring and Arnold W.M. Smeulders (2001). 'Filter image browsing: Interactive image retrieval by using database overviews'. *Multimedia Tools and Applications* **15**(1), 83–103.

Vesanto, J., Johan Himberg, Esa Alhoniemi and Juha Parhankangas (2000). Som toolbox for matlab 5. Report A57. Helsinki University of Technology, Neural Networks Research Centre. Espoo, Finland.

Vetterli, M. and J. Kovacevic (1995). *Wavelets and Subband Coding*. Prentice-Hall, Inc.

Vincent, L. and Pierre Soille (1991). 'Watersheds in digital spaces: An efficient algorithm based on immersion simulations'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**(6), 583–598.

Virage Inc. (1998). 'Virage Visual Information Retrieval Image Engine'. web page: `http://www.virage.com/market/vir.html`.

Vistex (1995). 'MIT media LAB's vision texture collection'. http://www-white.media.mit.edu/vismod/imagery/VisionTexture/vistex.html.

Voorhees, E. M. (Ed.) (2001). *The Tenth Text REtrieval Conference (TREC 2001)*. National Institute of Standards and Technology.

W3 Consortium (2002*a*). 'Hypertext transfer protocol'. Web page : `http://www.w3.org/Protocols/`.

W3 Consortium (2002*b*). 'Synchronized multimedia integration language'. Web page `http://www.w3.org/AudioVideo/`.

Wan, X., Zijun Yang and C.-C. Jay Kuo (1998). Efficient interactive image retrieval with multiple seed images. In C.-C. J. Kuo, S.-F. Chang and S. Panchanathan (Eds.). 'Multimedia Storage and Archiving Systems III (VV02)'. Vol. 3527 of *SPIE Proceedings*. Boston, Massachusetts, USA. pp. 13–24. (SPIE Symposium on Voice, Video and Data Communications).

Ware, C. (2000). *Information Visualization. Perception for Design*. Morgan Kaufmann. San Francisco.

Westerveld, T., D. Hiemstra and F De Jong (2000). Extracting bimodal representations for language-based image retrieval. In 'Proceedings of the Eurographics Workshop : Multimedia '99.'. Springer-Verlag. Vienna, Austria. pp. 33—42.

White, D. A. and Ramesh Jain (1996*a*). Algorithms and strategies for similarity retrieval. Technical Report VCL-96-101. Visual Computing Laboratory, University of California, San Diego. 9500 Gilman Drive, Mail Code 0407, La Jolla, CA 92093-0407.

White, D. A. and Ramesh Jain (1996*b*). Similarity indexing: algorithms and performance. In I. K. Sethi and R. C. Jain (Eds.). 'Storage and Retrieval for Still Image and Video Databases IV'. Vol. 2670 of *SPIE Proceedings*. pp. 62–73.

Wickerhauser, M. V. (1991). Fast approximate factor analysis. In 'SPIE Proceedings : Curves and Surfaces in Computer Vision and Graphics II'. pp. 23–32.

Wickerhuaser, M. V. (1994). *Adapted Wavelet Analysis from Theory to Software*. IEEE Press.

Wiemer-Hastings, P. M. (1999). How latent is latent semantic analysis?. In D. Thomas (Ed.). 'Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99-Vol2)'. Morgan Kaufmann Publishers. S.F.. pp. 932–941.

Winter, A. and Chahab Nastar (1999). Differential feature distribution maps for image segmentation and region queries in image databases. In 'IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL'99)'. Fort Collins, Colorado, USA. pp. 9–17.

Witten, I. H., Alistair Moffat and Timothy C. Bell (1999). *Managing gigabytes: compressing and indexing documents and images*. 2 edn. Morgan Kaufmann Publishers.

Witter, D. I. and Michael W. Berry (1998). 'Downdating the latent semantic indexing model for conceptual information retrieval'. *The Computer Journal* **41**(8), 589–601.

Wood, M. E., Neill W. Campbell and Barry T. Thomas (1998). Iterative refinement by relevance feedback in content-based digital image retrieval. In 'Proceedings of The Fifth ACM International Multimedia Conference (ACM Multimedia 98)'. Bristol, UK. pp. 13–20.

Woods, W. A. (1975). What's in a link? foundations in semantic networks. In D. G. Bobrow and A. M. Collins (Eds.). 'Representations and Understanding: Studies in Computer Science'. Academic Press. pp. 32–82.

Wu, P., B. S. Manjunath, S. D. Newsam and H. D. Shin (1999). A novel texture descriptor for image retrieval and browsing. In 'IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL'99)'. Fort Collins, Colorado, USA. pp. 3–7.

Yang, M.-H., D.J. Kriegman and N. Ahuja (2002). 'Detecting faces in images: A survey'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(1), 34 –58.

Yoshitaka, A. and T. Ichikawa. (1999). 'A survey on content-based retrieval for multimedia databases'. *IEEE Transactions on Knowledge and Data Engineering* **11**(1), 81–93.

Young, P. G. (1994). Cross-language information retrieval using latent semantic indexing. Technical Report UT-CS-94-259. Department of Computer Science, University of Tennessee. Thu, 19 Oct 100 17:31:51 GMT.

Zarita, R. and S. Lelandais (1997). Wavelets and high order statistics for texture classification. In M. Frydrych, J. Parkkinen and A. Visa (Eds.). 'The 10th Scandinavian Conference on Image Analysis (SCIA'97)'. Pattern Recognition Society of Finland. Lappeenranta, Finland. pp. 95–102.

Ze Wang, J., Gio Wiederhold, Oscar Firschein and Sha Xin Wei (1997). Wavelet-based image indexing techniques with partial sketch retrieval capability. In 'Proceedings of the Fourth Forum on Research and Technology Advances in Digital Libraries'. Washington D.C.. pp. 13–24.

Zha, H. and Horst D. Simon (2000). 'On updating problems in latent semantic indexing'. *SIAM Journal on Scientific Computing* **21**(2), 782–791.

Zha, H. and Zhenyue Zhang (2000). 'Matrices with low-rank-plus-shift structure: Partial SVD and latent semantic indexing'. *SIAM Journal on Matrix Analysis and Applications* **21**(2), 522–536.

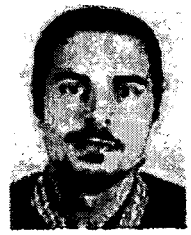
Curiculum

ZORAN PECENOVIC

Dipl. Ing. Computer Science, EPFL
Dr. in Computer Science & Communications, EPFL

Age: 28
Born in Zagreb, Croatia
Croatian and Yugoslav Nationality
B working permit

17, Avenue de Morges
1004 Lausanne,
Switzerland
Phone +41 076 411 11 74
Fax +41 021 625 82 40
E-mail: Zoran.Pecenovic@epfl.ch



Summary of qualifications

nov. 2002 Ecole Polytechnique Fédérale Lausanne, Switzerland
Doctor in Computer Science & Communications
Delivered by the Faculty of Computer Science and Communication Systems

oct. 1998 Ecole Polytechnique Fédérale Lausanne, Switzerland
Pre-doctoral school in Communication Systems

apr. 1997 Ecole Polytechnique Fédérale Lausanne, Switzerland
Bachelor's degree in Computer Science

Previous Education

1992 **Baccalauréat & Maturité Fédérale,** Neuchâtel, Switzerland

1989 **Elementary school ,** Berne, Switzerland

Professional experience

apr. 1997 – dec. 2002 Audio-Visual Communications Laboratory
Human Computer Interaction group of the Database Lab.
EPFL, Lausanne, Switzerland

Assistant - doctorant

Research & Development
Lecturing (Human Computer interaction, Digital Signal Processing, Programming courses),
Student project supervision,
doctoral dissertation.

Languages

English	Perfect oral and written (Cambridge Certificate of Proficiency in English)
French	Perfect oral and written, bilingual
German	Very good oral and good written knowledge Swiss education level
Italian	Perfect oral and good written knowledge
Serbo-Croatian	Mother-tongue
Spanish	Good oral knowledge

Objectives

To integrate a dynamic and motivated team, to fully benefit both the company's and my personal goals.

Achieve higher expertise in information storage, access, retrieval and visualization domains.

Gain experience in project management in industrial and economic domains.

Publications

- PECENOVIC, ZORAN. 1997. *Image retrieval using Latent Semantic indexing.* Final year graduate thesis, AudioVisual Communications Lab, Ecole Polytechnique Fédérale de Lausanne, Switzerland.
- PECENOVIC, ZORAN. 1998. *Finding Rainbows on the Internet.* M.Phil. thesis, Ecole Polytechnique Fédérale de Lausanne, Switzerland.
- PECENOVIC, ZORAN, DO, MINH, AYER, SERGE, & VETTERLI, MARTIN. 1998. "New methods for image retrieval". Pages 242-246 of: *Proceedings of the International Congress on Imaging Science*, vol. 2.
- PU, PEARL, & PECENOVIC, ZORAN. 2000. "Dynamic Overview Techniques for Image Retrieval". In: *VisSym'00, Second Joint Eurographics-IEEE - TCVG Symposium on Visualization.*
- PECENOVIC, ZORAN, DO, MINH, VETTERLI, MARTIN, & PU, PEARL. 2000. "Integrated Browsing and Searching of Large Image Collections". In: *International Conference on Visual Information Systems (Visual 2000).*
- MÜLLER, WOLFGANG, MÜLLER, HENNING, MARCHAND-MAILLET, STÉPHANE, PUN, THIERRY, SQUIRE, DAVID MCG., PECENOVIC, ZORAN, GIESS, CHRISTOPH, & DE VRIES, ARJEN P. 2000. "MRML: A Communication Protocol for Content-Based Image Retrieval". In: *International Conference on Visual Information Systems (Visual 2000).*
- MÜLLER, WOLFGANG, PECENOVIC, ZORAN, MÜLLER, HENNING, MARCHAND-MAILLET, STEPHANE, PUN, THIERRY, SQUIRE, DAVID, VRIES, ARJEN P. DE, & GIESS, CHRISTOPH. 2000. "MRML: An Extensible Communication Protocol for Interoperability and Benchmarking of Multimedia Information Retrieval Systems". In: *SPIE Photonics East – Voice, Video, and Data Communications.*
- PECENOVIC, ZORAN, AYER, SERGE, & VETTERLI, MARTIN. 2001. "Joint Textual and Visual Cues for Retrieving Images Using Latent Semantic Indexing". In: *Proceedings of the International Workshop on Content-Based Multimedia Indexing.*
- PECENOVIC, ZORAN. 2002. *Integrating visual and semantic descriptions for effective, flexible and user-friendly image retrieval*, Ph.D. Dissertation, Ecole Polytechnique Fédérale de Lausanne, Switzerland.

Expertise

Multi-media databases access and retrieval, Information retrieval, Data warehouses.

Digital Signal Processing, Statistical and factor analysis, Data-mining.

Human-computer Interaction, Usability testing.

Distributed computing, Communication protocols.

Development software lifecycles, UML.

Computer Literacy

Programming:	C/C++, Java, Perl; Python, Ada, Fortran; Tcl, Lisp, Pascal, Php, PL/SQL.
Database technology:	Oracle 8i, SQL Server, MySQL, ODBC/JDBC.
Network technology:	TCP/IP, HTML, XML, J2EE (Java Beans), JSP, ASP, PVM.
Operating systems:	Windows 95, 98, Me, NT, 2000; Solaris, Irix, Linux.
Other:	Matlab, MS Visual C++, Forte, Vtk, Office, File Maker, Lotus Notes, Adobe, Lightwave.

Interests and activities

Literature, art, music, travelling, basketball, skiing, hiking, photography.

Projects

- 1998-2002: Individual Ph.D. project on multimedia retrieval. Complete system from scratch using true multimedia approach and advanced visualization techniques.
- 1998-2002: Multimedia retrieval: indexing, user interaction and communication issues. Collaborative Project financed in part by the Swiss National Fund for Scientific Research.
- 1998-2002: Management of several student semester projects in Digital Signal Processing, Networking, Database, Intranet & Internet applications.
- 1998: Information theory and wavelet packet expansions applied to image retrieval. Pre-doctoral school semester project.
- 1997: Gigabyte Image viewer. Interactive system for smooth navigation in huge images, with semantic and cartographic registration and integration.
- 1997: Image retrieval using Latent Semantic Indexing. Image processing, database management, user interaction. Individual final year diploma project.
- 1996: Implementation of a user interface for multimedia retrieval on the O2 object oriented database management system. Individual semester project.
- 1995: Implementation of the Parallel Virtual Machine on a Transputer based parallel computer. Individual semester project.
- 1994: Distributed implementation of an interactive game. Group project for software engineering class.

References

Prof. Martin Vetterli,
head of the Audio Visual Communications Laboratory

LCAV / I&C, EPFL,
1015 Lausanne
Tel: 021 693 5698
Email: Martin.Vetterli@epfl.ch

Dr. George Melissargos,
Technical marketing manager, collaborative visualization EMEA Silicon Graphics,

Av. Louis-Casaï 18
1209 Genève
Tel: 078 6805332
Email: gmelissa@sgi.com

Prof. Stefano Spaccapietra,
head of the Database Laboratory

LBD / I&C, EPFL,
1015 Lausanne
Tel: 021 693 5210
Email: Stefano.Spaccapietra@epfl.ch

Dr. Laurent Balmelli,
researcher at IBM T. J. Watson, NY center

IBM T. J. Watson Research Center
30 Saw Mill River Road (Route 9A)
Hawthorne, NY 10532
Email: balmelli@watson.ibm.com

Dr. Pearl Pu-Faltings,
head of the Human Computer Interaction Research Group/ LBD

LBD / I&C, EPFL,
1015 Lausanne
Tel: 021 693 6081
Email: Pearl.Pu@epfl.ch

Sandro Caliz
project manager / consultant at Cambridge Technology Partners

Cambridge Technology Partners,
Air Center, 16, ch. des Coquelicots
CH-1214 Vernier
Tel: 076 428 9513

Email: Sandro.Caliz@ctp.com